

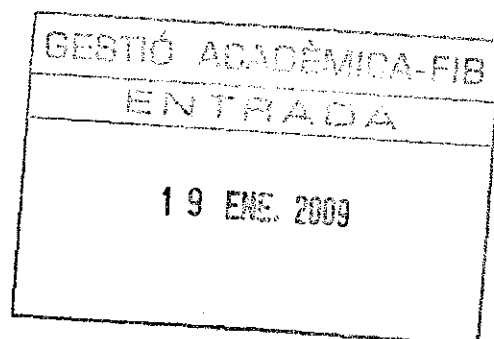
Títol: Estudi de noves plataformes robòtiques assistents en cirurgia laparoscòpica

Volum: 1

Alumne: Rubén Abad Prados

Director/Ponent: Josep Amat Girbau

Departament: ESAII



*Als meus pares, que es van preocupar de tota la resta per que jo només em
preocupes per estudiar i trobar el meu camí a la vida.*

Als meus professors, per contagiar-me el seu esperit rigorós i somiador.

*Als meus amics, per estar al meu costat quan la pressió de la carrera
m'ofegava.*

INTRODUCCIÓ.....	11
SISTEMA ROBÒTIC PER CIRURGIA LAPAROSCÒPICA	15
INTRODUCCIÓ.....	15
HISTORIA.....	16
PROPOSTA DEL SISTEMA	19
REQUERIMENTS DEL SISTEMA.....	21
DIAGRAMA GENERAL	22
MÒDULS IMPLICATS	25
<i>Modelat 3D.....</i>	<i>25</i>
<i>Simulador 3D.....</i>	<i>26</i>
<i>Màster.....</i>	<i>26</i>
<i>Estratègia.....</i>	<i>27</i>
<i>Unitat de control.....</i>	<i>27</i>
METODOLOGIA DE DISSENY I DESENVOLUPAMENT	28
<i>Proposta d'arquitectura i modelat 3D.....</i>	<i>28</i>
<i>Simulació i validació de l'arquitectura.....</i>	<i>31</i>
<i>Disseny del Prototip.....</i>	<i>33</i>
PLANIFICACIÓ.....	33
CONCLUSIONS	35
CINEMÀTICA PER AL ROBOT DE 3 BRAÇOS.....	37
INTRODUCCIÓ	37
CINEMÀTICA DIRECTA (DK).....	38
<i>Resolució mitjançant mètodes geomètrics directes.....</i>	<i>39</i>
<i>Resolució mitjançant matrius homogènies.....</i>	<i>39</i>
<i>Resolució mitjançant els paràmetres de Denavit-Hartenberg.....</i>	<i>42</i>
<i>Resolució mitjançant quaternions.....</i>	<i>44</i>
<i>Resolució de la cinemàtica directa del manipulador de 3 braços</i>	<i>45</i>
CINEMÀTICA INVERSA (IK)	48
<i>Resolució mitjançant mètodes geomètrics directes.....</i>	<i>49</i>
<i>Resolució a partir de la matriu de transformació homogènia.....</i>	<i>50</i>
<i>Resolució mitjançant mètodes numèrics.....</i>	<i>52</i>
<i>Resolució de la cinemàtica inversa del manipulador de 3 braços.....</i>	<i>53</i>
CONCLUSIONS	58
SOFTWARE DE SIMULACIÓ INTERACTIVA PER AL ROBOT PÒRTIC DE 3 BRAÇOS.....	59
INTRODUCCIÓ.....	59
ARQUITECTURA: DIAGRAMES DE CLASSES I SISTEMES IMPLICATS	60
SISTEMA 3D	62
SISTEMA DE REPRESENTACIÓ.....	64
SISTEMA RPQ	65
SISTEMA D'E/S.....	66
SISTEMA SÈRIE	67
CONCLUSIONS	67
SEGUIMENT INTEL·LIGENT DE LES EINES.....	69
INTRODUCCIÓ.....	69
ESTRATÈGIA DE SEGUIMENT	71
EL SIMULADOR D'UN BRAÇ.....	74
IMPLEMENTACIÓ I PROVES.....	76
CONCLUSIONS	78
RESULTATS FINALS.....	81
INTRODUCCIÓ.....	81
AVALUACIÓ TÈCNICA	82
<i>Estudi de les mides inicials del robot.....</i>	<i>82</i>
<i>Estudi de la precisió de la cinemàtica</i>	<i>83</i>

AVALUACIÓ MÈDICA	88
NOVES PROPOSTES D'INVESTIGACIÓ.....	89
CONCLUSIONS:	91
PRESSUPOST DEL SISTEMA	93
INTRODUCCIÓ	93
DESENVOLUPAMENT DEL PROTOTIP	93
ESTUDI DE RENDIBILITAT	94
PRESSUPOST DE PRODUCCIÓ	96
CONCLUSIONS	97
BIBLIOGRAFIA	99
LLIBRES DE CONSULTA	99
WEBS DE CONSULTA	99
ANNEX I: EINES PER LA IMPORTACIÓ/EXPORTACIÓ DE MODELS GEOMÈTRICS DES DEL SOFTWARE DE DISSENY AL SOFTWARE DE SIMULACIÓ	101
INTRODUCCIÓ:	101
ESPECIFICACIÓ DEL FORMAT RPQ (.TRIS)	102
<i>FIGURA A1.2: EXTRACTE D'UN ARXIU .TRIS</i>	103
EINES	103
FACTOR DE CONVERSIÓ	104
COORDENADES GLOBALS I LOCALS	104
IMPORTADOR/EXPORTADOR .TRIS	104
EL FORMAT DIMUC	105
<i>Justificació</i>	105
<i>Especificació</i>	105
IMPORTADOR/EXPORTADOR .DIMUC	107
EL CONVERSOR DIMUC2TRIS	107
NOVETATS DE LA VERSIÓ 2.0	108
NOVETATS DE LA VERSIÓ 3.0	109
INSTAL·LACIÓ	109
VERSIONS:	110
ANNEX II: MANUAL DEL SOFTWARE DE CONTROL DEL ROBOT PÒRTIC D'UN BRAC	113
INTRODUCCIÓ:	113
L'ENTORN DE SIMULACIÓ	114
OPERACIONS PRINCIPALS:	115
CONNEXIÓ AMB EL ROBOT	116
INPUT DEL ROBOT I COMPORTAMENT	117
CAL·LIBRACIÓ DELS SENSORS DE POSICIÓ	118
MEMORITZACIÓ I REPRODUCCIÓ DE POSICIONS ESPECÍFIQUES	121
CONFIGURACIÓ DE L'ESTRATÈGIA DE SEGUIMENT	122
CINEMÀTICA	123
<i>Cinemàtica Directa</i>	123
<i>Cinemàtica Inversa</i>	124
DRECCERES DE TECLAT	125
ANNEX III: PLÀNOLS DEL PROTOTIP	127

Capítol 1:

Introducció

La cirurgia endoscòpica és una tècnica quirúrgica que es practica introduint les eines dintre del pacient i manegant-les des de. Amb l'ajut d'una càmera el cirurgià pot veure el camp operatori. Aquestes tècniques són mínimament invasives ja que no calen grans talls amb el bisturí com s'haurien de fer en cirurgia oberta o tradicional amb la qual cosa el període postoperatori és més ràpid i confortable.

Les seves aplicacions cada cop són més nombroses a causa dels beneficis abans esmentats. Pot ser utilitzada en cirurgia abdominal, articular, ginecologia, toràcica, etc. En el cas de la endoscòpia abdominal o laparoscòpia es necessari fer altres talls petits per introduir-hi l'instrumental específic. Aquest són versions especialitzades dels instruments d'us comú en les intervencions habituals, com pinces, tisores, separadors, agulles per cosir, etc, però més llargs i fins per poder ser emprats amb comoditat per forats petits.

La cirurgia laparoscòpica es realitza gràcies a una vídeo càmera i un conjunt d'òptiques que s'introdueix dintre del cos a traves d'una incisió (figura 1.1). Una font de llum freda s'encarrega de donar la llum necessària per il·luminar el camp operatori. Aquesta llum és introduïda dintre del pacient mitjançant fibra òptica.

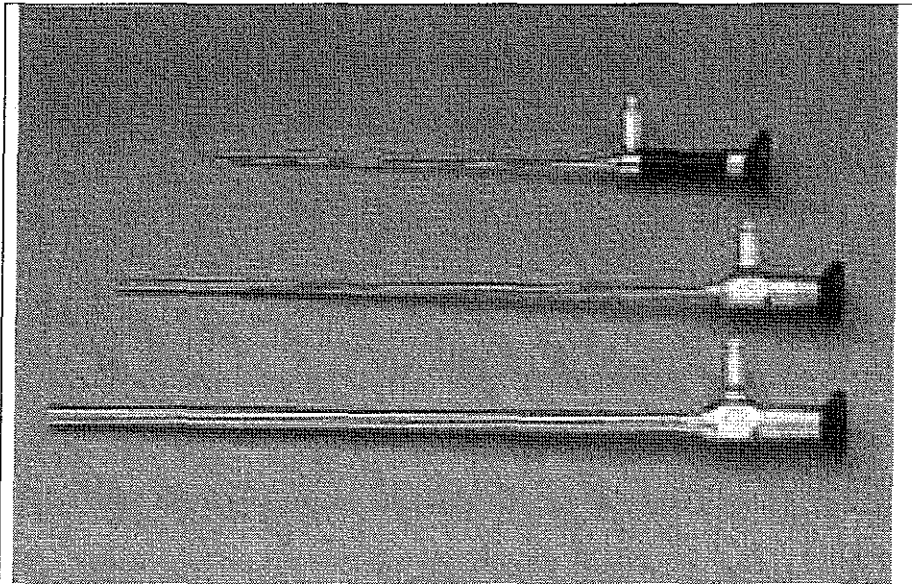


Figura 1.1: Diferents òptiques emprades en intervencions per laparoscòpia.

L'equip laparoscòpic en el quiròfan inclou món itors d'alta resolució on el cirurgià y el seu equip poden veure les imatges produïdes per la vídeo càmera. Això permet veure amb gran detall petites àrees de treball.

Aquestes eines no entren directament friccionant contra la pell sinó que primer es col·loca un cilindre per on l'eina entra i surt, el *froccar*. A més el punt de fulcre es fa servir com a punt de pivot per desplaçar les eines cap els costats.

El fet de moure l'eina pivotant sobre un punt te dos greus inconvenients. El primer d'ells es que els moviments que fa el cirurgià sobre l'eina s'inverteixen a l'extrem, per exemple si es vol desplaçar l'extrem de l'eina a l'esquerra el cirurgià ha de moure el mànec cap a la dreta. L'altre inconvenient es que el desplaçament que fa el cirurgià no es el mateix que es fa a l'extrem sinó que es proporcional a la part de l'eina que hi ha dintre del pacient. Si el cirurgià ha introduït molt l'eina un petit moviment de la seva mà es converteix en un gran moviment a

l'extrem. Actualment aquest inconvenients es solucionen gràcies a l'habilitat del cirurgià.

Un altre inconvenient d'aquest procediment es el fet de que ha d'haver un assistent encarregant-se de guiar la càmera per tant de focalitzar la imatge sobre l'àrea d'interès dins del camp operatori o on el cirurgià precisi. Aquest assistent no només ha de portar la càmera sinó que s'ha de col·locar en llocs del quiròfan on no dificulti l'activitat dels cirurgians. A vegades certs moviment de la càmera implica que l'assistent ha de vorejar tota la taula d'operacions i trobar un nou lloc de treball.

El fet de treballar a través d'una càmera comporta altres problemes. Aquesta s'ha de moure allà on el cirurgià requereixi i, un cop arribada a la seva posició s'ha de mantenir estable durant tota la intervenció. La medicina moderna està optant pels robots a l'hora de guiar les càmeres ja que són més petits i es poden situar de forma que gairebé no molestin al cirurgià, a més de ser més estables i precisos que una persona.

La introducció de noves plataformes quirúrgiques assistides per sistemes robòtics és objecte d'estudi de moltes empreses i organismes públics. L'objectiu principal és aconseguir millors eines i procediment mèdics per millorar l'eficiència dels sistemes sanitaris actuals.

Des de la Fundació Parc Taulí conjuntament amb el departament d'*Enginyeria de Sistemes, Automàtica i Informàtica Industrial* (ESAI) de la Universitat Politècnica de Catalunya (UPC) s'estan desenvolupant noves tècniques i eines per cirurgia laparoscòpica assistida per sistemes robòtics.

Capítol 2:

Sistema robòtic per cirurgia laparoscòpica

Introducció

Sovint els avenços en tècniques quirúrgiques més complexes compten amb l'inconvenient de que depenen, en gran mesura, de la destresa del cirurgià. Aquest factor humà limita molt la pràctica de noves intervencions a nivell teòric. La introducció de robots, molt més precisos i maniobrables que les persones ha obert nous camps d'investigació i actuació com són les tècniques basades en microcirurgia.

La cirurgia assistida amb robots separa i delega les tasques a aquells que estan millor preparats per realitzar-les. Amb aquest nou paradigma el cirurgià s'encarrega de veure *què* i *com* s'ha de practicar la intervenció i el robot s'encarrega d'executar-la.

Un clar exemple es troba en les intervencions de còrnia, on el metge calcula el tall que s'ha de fer per corregir la deformació ocular i el robot s'encarrega de realitzar-lo amb una precisió de l'ordre de la micra

En el camp de la cirurgia mínimament invasiva i, concretament les intervencions per cirurgia laparoscòpia el referent més important al mercat és el robot *Da Vinci* (<http://www.intuitivesurgical.com/index.aspx>), un robot teleoperat, de 4 braços preparat per portar la càmera i diverses eines. *Da Vinci* incorpora un sistema de visió estereoscòpica, de tal forma que el metge té una sensació quasi real d'estar treballant a

dintre del pacient (per la percepció de profunditat) en comptes de treballar mirant una imatge 2D.

Aquest avenços fan de la teleoperació una realitat cada cop més propera que permetrà descentralitzar els pacients dels cirurgians reduint els costos de les intervencions i proporcionant un servei immediat i de qualitat als pacients.

Un altre exemple de com aprofitar els avantatges de la robòtica va ser el *Robodoc* (<http://www.robodoc.com/>), un robot dissenyat per tallar l'ós en una implantació de pròtesis per que aquesta encaixi perfectament. La precisió que s'aconseguia amb aquest robot era de l'ordre de 10 vegades més gran que amb el procediment manual.

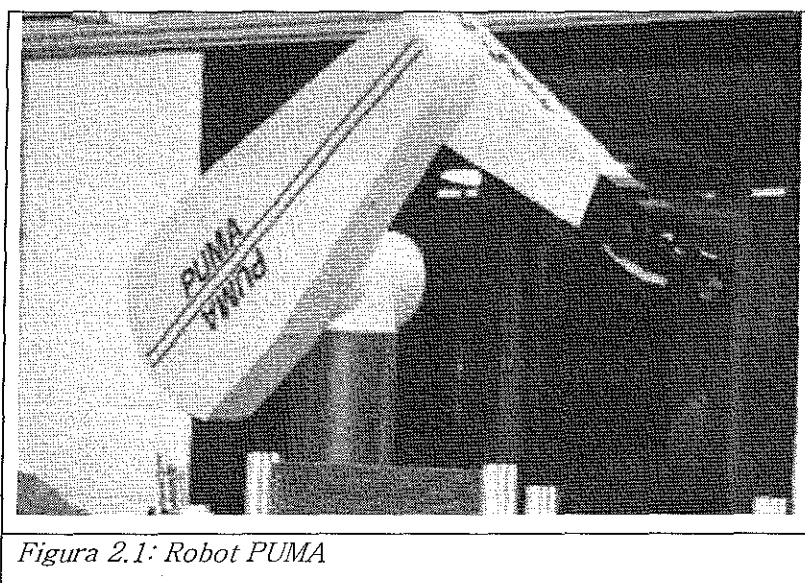
La robòtica també s'ha introduït en els processos de rehabilitació on els pacients han de realitzar un seguit de moviments món òtons i repetitius, tasca que els robots porten dècades fent a les cadenes de muntatge industrials. Aquest robots donen la possibilitat als pacients de fer la rehabilitació sense la intervenció directa d'un fisioterapeuta. Si l'execució dels exercicis recau sota la responsabilitat d'un robot els fisioterapeutes poden centrar la seva feina en el seguiment de més pacients. Amb això s'aconsegueix donar un millor servei a més gent i reduint el cost del desplaçament i hores dedicats per l'especialista a l'execució dels exercicis de rehabilitació.

Història

Un dels grans avenços a la robòtica va ser el disseny d'un braç articulat per part de l'enginyer mecànic *Victor Scheinman* a la universitat

de Stanford (Califòrnia) conegut com a *Braç Manipulador Universal programable* (PUMA) (figura 2.1) l'any 1975. Molts dels manipuladors posteriors es basen en aquest disseny.

Deu anys més tard es fa la primera operació quirúrgica on intervé un robot. Es va emprar un PUMA 560 per col·locar una agulla en una biòpsia cerebral. Els bons resultats de la intervenció van mostrar l'enorme potencial que podia aportar la robòtica a la cirurgia.



L'estudi d'aquestes noves opcions va trigar en donar fruit. Al 1993 apareix l'*AESOP 1000* un robot per guiar la càmera en una operació per laparoscòpia. El metge interactuava amb el robot amb comandes per veu. No és fins al 1997-1998 que apareixen els primers robots per telepresència i teleoperació : *ZEUS* (*Computer Motion*) (figura 2.2) i *Da Vinci* (*Intuitive Surgical*) (figura 2.3).

ZEUS és un sistema de 3 braços amb una càmera *AESOP* i una consola on el metge manipula els braços mirant a una imatge 2D.

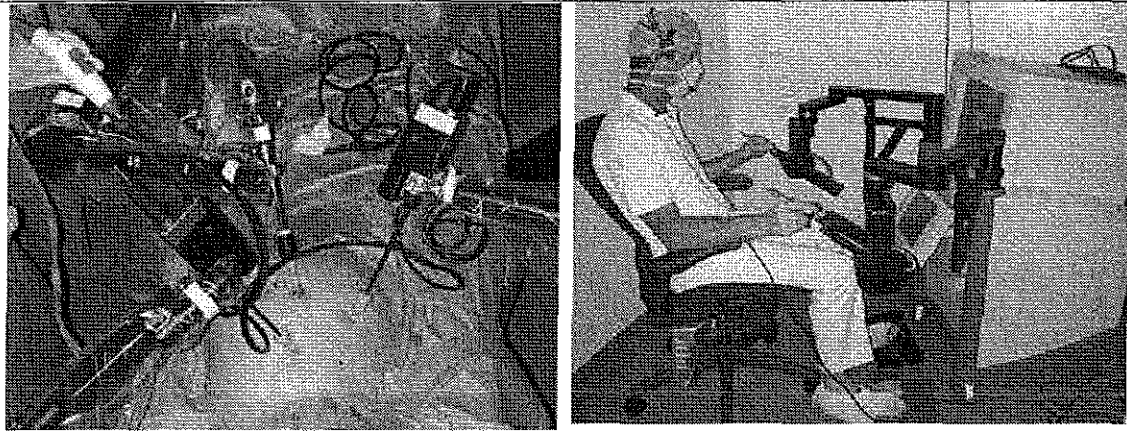


Figura 2.2: Sistema Zeus

L'any 2001 es realitza amb èxit la primera teleoperació transatlàntica amb el sistema *ZEUS*, una colecistectomia laparoscòpica de Nova York a Estrasburg. Es va emprar una comunicació per fibra òptica de tal forma que el retard del senyal era de 150 ms tot i haver 7.500 km. de distància. Actualment no es comercialitza i el suport pels que queden operatius el dona l'empresa *Intuitive Surgical*.

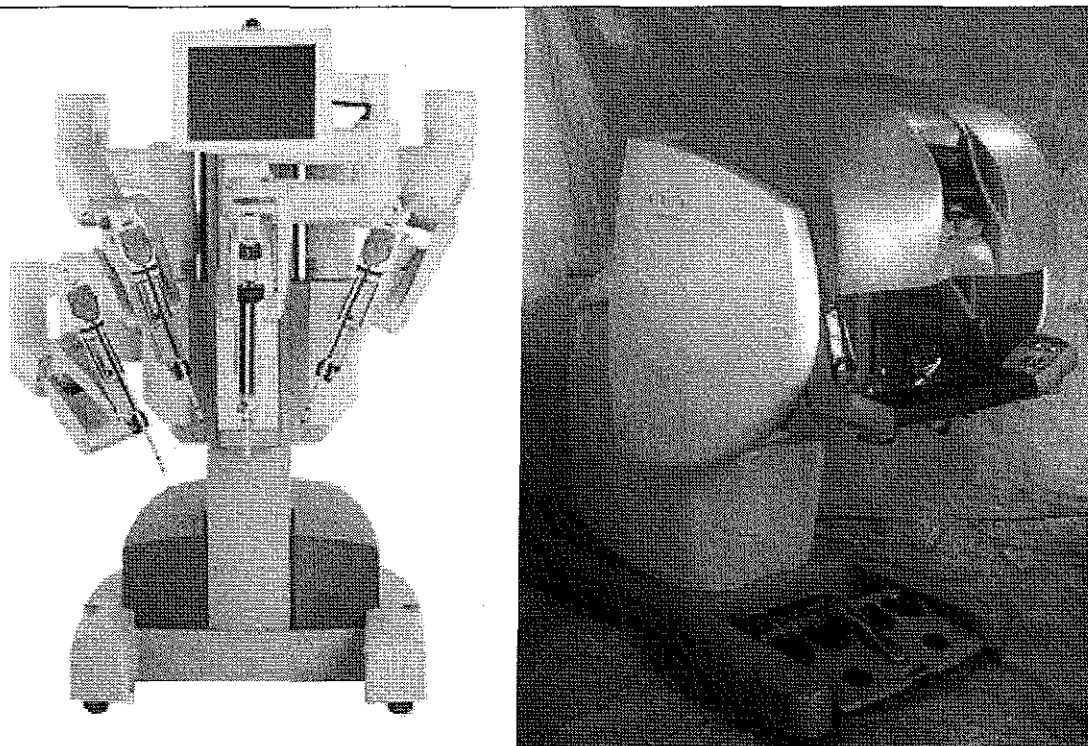


Figura 2.3: Sistema Da Vinci

Per la seva part *Da Vinci* incorpora 4 braços en la seva darrera versió. La consola on treballa el metge proporciona imatge 3D per estereovisió i el control de les eines es produeix mitjançant un sistema hàptic.

Da Vinci és el sistema de referència avui dia. Tot i això ha començat a mostrar les seves febleses. A mesura que les intervencions cooperatives amb robots són més nombroses els metges reclamen més braços per operar. En el sistema *Da Vinci* els braços surten d'un mateix suport central, la qual cosa fa que afegir-ne més sigui complicat per interferències a l'espai de treball. Noves vies d'investigació plantegen fer sistemes amb robots independents de pocs braços de forma que es puguin distribuir millor sobre el camp operatori (robots cooperatius).

Moltes entitats, tant públiques com privades continuen treballant en noves incorporacions de la robòtica al món de la medicina. Les intervencions d'alt risc o intervencions de microcirurgia són les principals candidates a la incorporació de robots que proporcionen eines reduïdes, més precisió i millor maniobrabilitat.

Proposta del sistema

La *Fundació Parc Taulí* conjuntament amb el departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial (ESAI) de la *Universitat Politècnica de Catalunya (UPC)* estan preparant com a projecte un sistema de cirurgia laparoscòpica assistida.

Una intervenció tradicional implica que el metge ha d'invertir els moviments que fa en agafar l'eina per un extrem per que faci el

moviment desitjat a l'extrem que hi ha dintre del pacient. El sistema proposa una solució per fer de les intervencions un procés més intuïtiu pel metge.

El sistema global estarà compost per un robot de 3 braços: un per portar la càmera i els altres dos per portar eines. També s'inclou el software de control i simulació i un conjunt de dispositius per a donar comandes al robot.

La novetat principal respecte als sistemes comercials actuals és el fet de proposar un sistema fàcilment portable, que no requereixi de cap equipament addicional a la sala d'operacions. D'aquesta forma el sistema es pot traslladar fàcilment entre quiròfans i guardar-se on no destorbi quan no es faci servir.

A més el sistema és adaptable a l'hora de donar comandes al robot per moure les eines. Mitjançant una capa d'abstracció de dispositius el sistema pot ser ampliat fàcilment per suportar diferents formes d'entrada de dades i comandes: joysticks, sistemes de força, sistemes de posició magnètics, sistemes de captura de moviment, etc...

A partir de la posició actual dels braços que porten les eines es calcula quin ha de ser el moviment que han de fer els robots per arribar a la posició desitjada. Un cop calculat el moviment es comprova si hi hauran col·lisions o algun altre problema que invalidi el nou desplaçament. Finalment, si el moviment es factible, els robots portaran les eines allà on el metge requereixi (figura 2.4).

El robot incorpora un braç dedicat a portar la càmera. Aquesta està pensada per ser portada de forma autònoma fent un seguiment de les

eines. Quan les eines es mouen per arribar a la zona de treball del cirurgià el sistema calcula quina és la posició on s'ha de portar la càmera, respectant una distància prefixada. Un cop arribat al lloc desitjat, la càmera es pot bloquejar o definir un radi d'histeresi. Si les eines no surten d'aquest radi la càmera no es mourà.

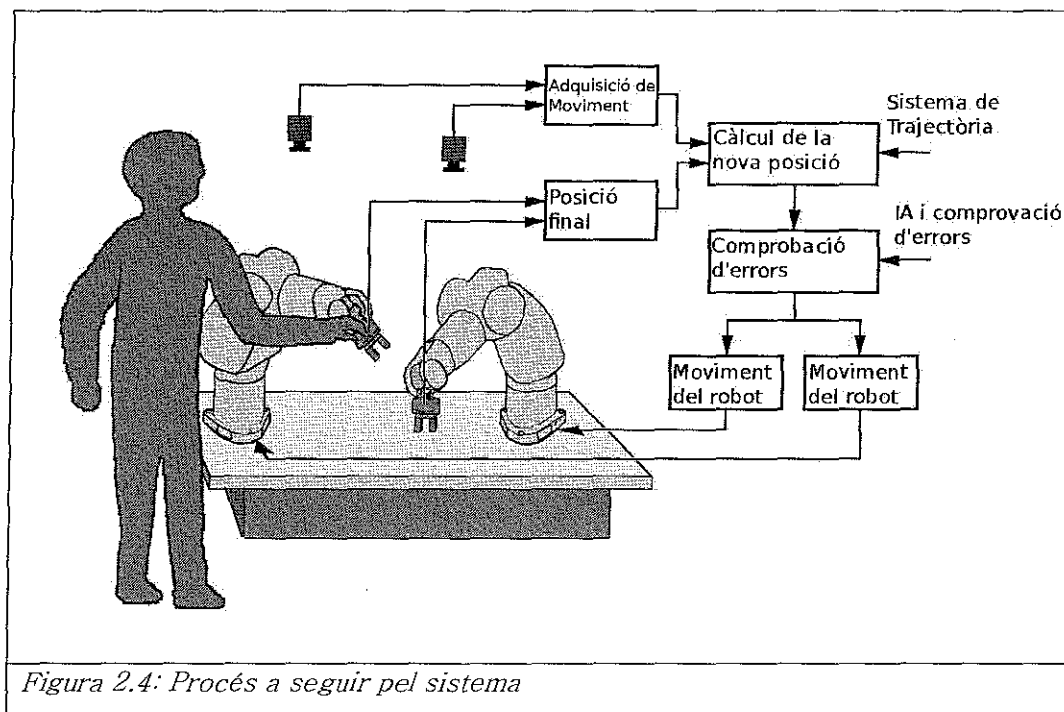


Figura 2.4: Procés a seguir pel sistema

Requeriments del sistema

Per a ser una solució vàlida el sistema ha de complir els següents requeriments:

1. Zona de treball: Els tres braços han de poder treballar sobre la zona abdominal d'un pacient. Les mides exactes d'aquesta zona les ha de determinar el departament mèdic com resultat d'un estudi antropomètric.
2. Ergonomia: Certes mides del robot s'han de poder ajustar a la mesura del pacient, per exemple l'alçada del pòrtic. Aquests

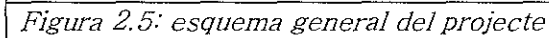
paràmetres, com els seus rangs de mobilitat s'han d'extreure a partir de l'estudi antropomètric anterior.

3. Mobilitat: El sistema no ha de requerir de cap modificació addicional de la sala d'operacions i s'ha de poder adaptar fàcilment a l'equipament mèdic estàndard disponible al mercat.
4. Interferència entre braços mínima: Els braços han de poder treballar sense que col·lisionin entre ells. El grau de compliment d'aquest requeriment determinarà, en gran mesura els tipus d'intervenció que es podran dur a terme amb el sistema.
5. Compatibilitat amb persones: La disposició del sistema ha de permetre que altres persones puguin intervenir a la operació. Principalment els braços robots s'han de situar on no destorbin a la resta de personal.
6. Precisió: Per la naturalesa de les intervencions per laparoscòpia la precisió dels moviments hauria d'estar entorn del mig mil·límetre.

Diagrama general

El conjunt total és un sistema retroalimentat per moure i fer actuar els braços conforme la voluntat del metge. A grans trets els braços es poden moure guiant-los manualment (pel que disposen d'un sensor de força), controlats per joystick o sensor de posició (útil per teleoperació) o seguint els moviments del metge gràcies a un dispositiu d'infrarojos

(telepresència). Amb això es veu que el diagrama principal de l'aplicació segueix el següent esquema (figura 2.5).



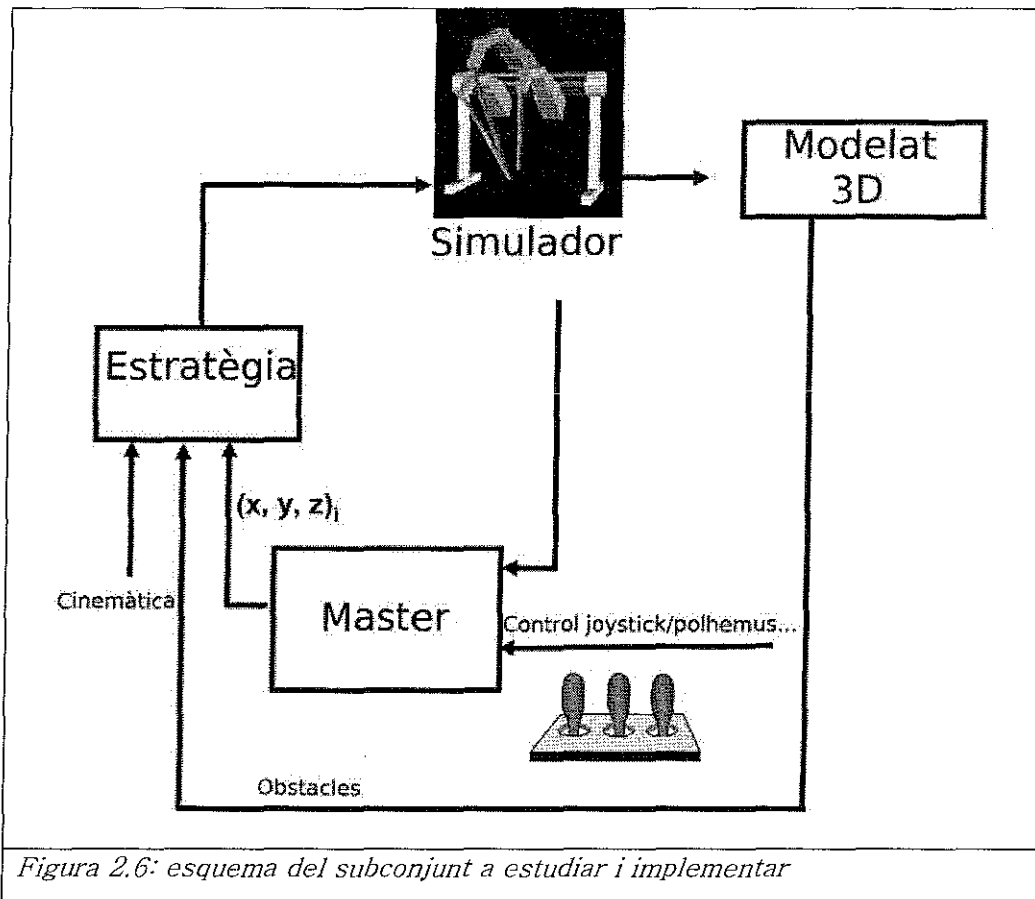
1. Adquirir les dades dels sensor per obtenir la configuració actual del robot i actualitzar els models 3D del simulador.
2. Esperar una nova ordre de moviment que pot venir directament del simulador, del sensor de força del braç, d'un sensor de posició, del sistema de visió o d'un comandament.
3. El màster processa les dades, filtra sorolls, ignora les dades que no estan dintre d'un cert llindar i fa correccions i adaptacions de les dades

4. Calcular les noves configuracions dels braços i la càmera, es verifiquen possibles col·lisions i d'altres restriccions. Si els moviments del robot són vàlids s'apliquen les noves configuracions.

Resumint, les tasques del sistema segueixen un esquema bàsic d'adquisició, tractament de dades i actualització de la configuració del robot.

L'abast d'aquest projecte inclou només l'estudi i software relatiu al modelat i simulador 3D, estratègia i Màster amb suport per joystick i per un sistema de posicionament magnètic (*Polhemus*). Aquests mòduls del sistema són els necessaris per estudiar la viabilitat del projecte. El simulador 3D permet substituir el robot real per estudiar-ne el seu comportament i provar les estratègies. El sistema de visió per seguiment dels moviments del cirurgià és un sistema d'entrada de dades a part que pot ser reemplaçat fàcilment per un sistema de posicionament magnètic (figura 2.6).

S'ha de tenir en compte que aquest subconjunt és principalment software i dispositius d'entrada de baix cost. Això permet estudiar la viabilitat del projecte sense haver d'invertir una bona part del pressupost (en el robot real o les càmeres, per exemple) i destinar aquests diners a la fabricació del prototip i la resta dispositius d'entrada/sortida. D'aquesta forma, la inversió en la fabricació del prototip robòtic es troba avalada.



Mòduls implicats

A continuació s'explicarà, amb més detall els mòduls implicats en el sistema general.

Modelat 3D

Aquest és el mòdul responsable de gestionar la informació relativa al disseny del robot. Per tenir aquest mòdul s'han d'obtenir els models 3D del robot a partir del disseny de l'arquitectura i fent servir un programa de modelat 3D. Juntament amb les taules de Denavit-Hartenberg, obtingudes de l'estudi cinemàtic, es podrà mostrar el model correctament i animar-lo.

Probablement serà necessari crear un conjunt d'eines per exportar els fitxers dels models 3D a un format més còmode per representar-los.

Simulador 3D

Un cop es tenen els models 3D i la informació de les taules de Denavit-Hartenberg es desenvolupa un simulador que mostrarà aquesta informació i la seva evolució. Aquest simulador mostrarà l'escena en diverses perspectives, incloent-hi una perspectiva amb càmera lliure i una altre simulant el punt de vista de la càmera laparoscòpica.

El simulador ha d'actualitzar la seva informació a mostrar llegint l'estat actual del robot de la unitat de control.

Màster

El Màster inclou, a més del sistema físic de comanda el sistema d'adquisició i filtrat de les dades obtingudes dels dispositius d'entrada que permeten la interacció amb el robot. Les dades estan pensades per tractar-se de forma incremental, de tal forma que el que processa el Màster són increments i variacions tant siguin del sensor de força, del joystick o dels sistemes de posicionament tan magnètics (*Polhemus*) com de visió. D'aquesta forma es més fàcil ignorar errors dels sensors ja que es poden configurar uns valors mínims i màxims d'increments acceptats. Això dóna més robustesa al sistema ja que no permet moviments i accions que puguin produir danys al pacient. S'ha de tenir en compte que quan un cirurgià opera per laparoscòpia els moviments que es realitzen dintre del pacient són moviments relativament petits.

Estratègia

El mòdul d'estratègia és l'encarregat de calcular la nova configuració que ha d'adoptar el robot per fer el moviment desitjat pel metge. A partir de la posició P_i actual dels braços amb les eines i l'increment de posició Δp_i es calcula la posició futura.

$$P_{i+1} = P_i + \Delta p_i$$

Mitjançant cinemàtica inversa es calcula la nova configuració per assolir la nova posició. Si hi ha diverses solucions s'escull la que impliqui un menor desplaçament del robot. Un cop se sap la configuració futura es comprova que aquesta nova configuració no faci tocar els braços.

Si la configuració dels braços amb eina es correcta es calcula la configuració que ha de prendre el braç de la càmera per donar un bon camp visual, respectant una distància prefixada i sense tocar amb els altres dos braços.

Quan se sap la configuració final de tots tres braços s'envien a la unitat de control per que mogui el robot.

Unitat de control

Aquest és un mòdul purament hardware microprogramat. La seva tasca és alimentar i fer moure els motors del robot segons una

configuració introduïda. També ha de permetre ser enquestada per actualitzar les dades del simulador.

De cara a la producció final aquest és un dels mòduls que haurà de complir més requisits per passar tots els estàndards clínics. Per exemple, el seu microcontrolador ha d'estar certificat per ús mèdic.

Metodologia de disseny i desenvolupament

Per portar a terme l'estudi de la viabilitat del sistema s'ha especificat un procés en tres fases, cada una d'elles amb unes fites ben diferenciades i fàcils d'assolir per garantir una mínima propagació d'errors a la fase següent.

Proposta d'arquitectura i modelat 3D

El sistema de control proposat té l'avantatge de ser independent dels robots que es facin servir sempre i quan el seu marge de maniobrabilitat estigui dintre d'uns rangs raonables. Tot i així la *Fundació Parc Taulí* va proposar un disseny d'un robot de tres braços pensat per situar-se en una estructura en pont sobre del pacient i fixable a la taula d'operacions. El principal avantatge d'aquesta proposta és que gairebé tota la zona de treball del robot s'ubica sobre l'àrea de treball on opera el cirurgià, a més de que així el robot es situa en una posició on no molestarà als cirurgians com ho faria si els braços es situessin als costats del pacient.

A l'hora de definir la mobilitat es va considerar convenient que els braços aprofitessin el pont per a moure's, d'aquesta forma el pont es converteix en una guia comuna als 3 braços que els permet moure's de forma lineal sobre un eix. D'aquesta forma els robots ja tenen un grau de llibertat.

Per a definir els altres dos graus es van plantejar dues propostes: dues articulacions rotacionals (a mode de colze) o una rotacional i una lineal (figura 2.7). Al final es va concloure per una rotacional i una lineal per dos motius:

1. L'estructura en colze pot ocupar massa espai per damunt del pacient, interferint amb les llampares del quiròfan.
2. La U.P.C. té la patent d'una arquitectura d'articulació lineal i modular que aprofita molt bé l'espai.

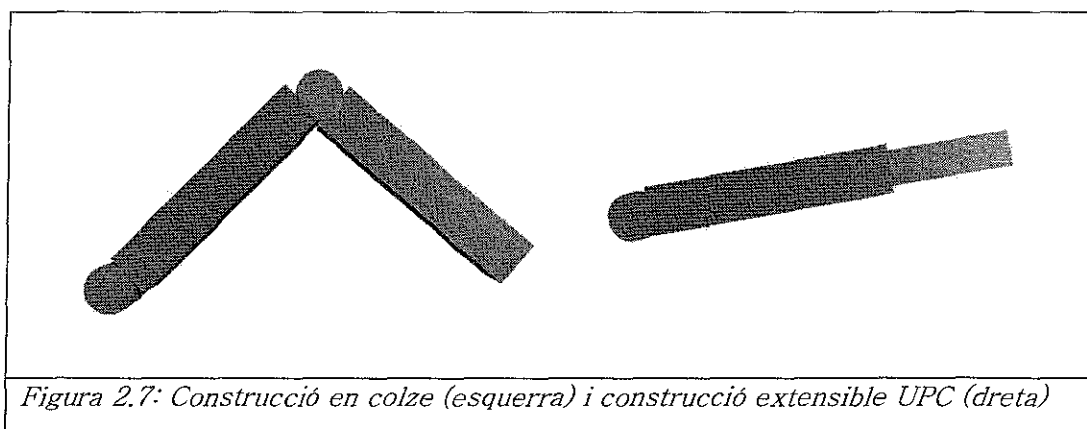


Figura 2.7: Construcció en colze (esquerra) i construcció extensible UPC (dreta)

Finalment, entre l'extrem del braç i el suport de l'eina es col·loquen dos graus de llibertat lliures per a permetre orientar-la. El resultat final es pot veure a la figura 2.8.

Les mides de la zona de treball del robot les ha de definir l'equip mèdic del projecte. Aquestes mides han de ser el resultat d'un estudi

antropomètric. D'aquest estudi també s'ha de concloure quines seran les parts del robot que seran ajustables, i en quina mesura per motius d'ergonomia i adaptabilitat al pacient.

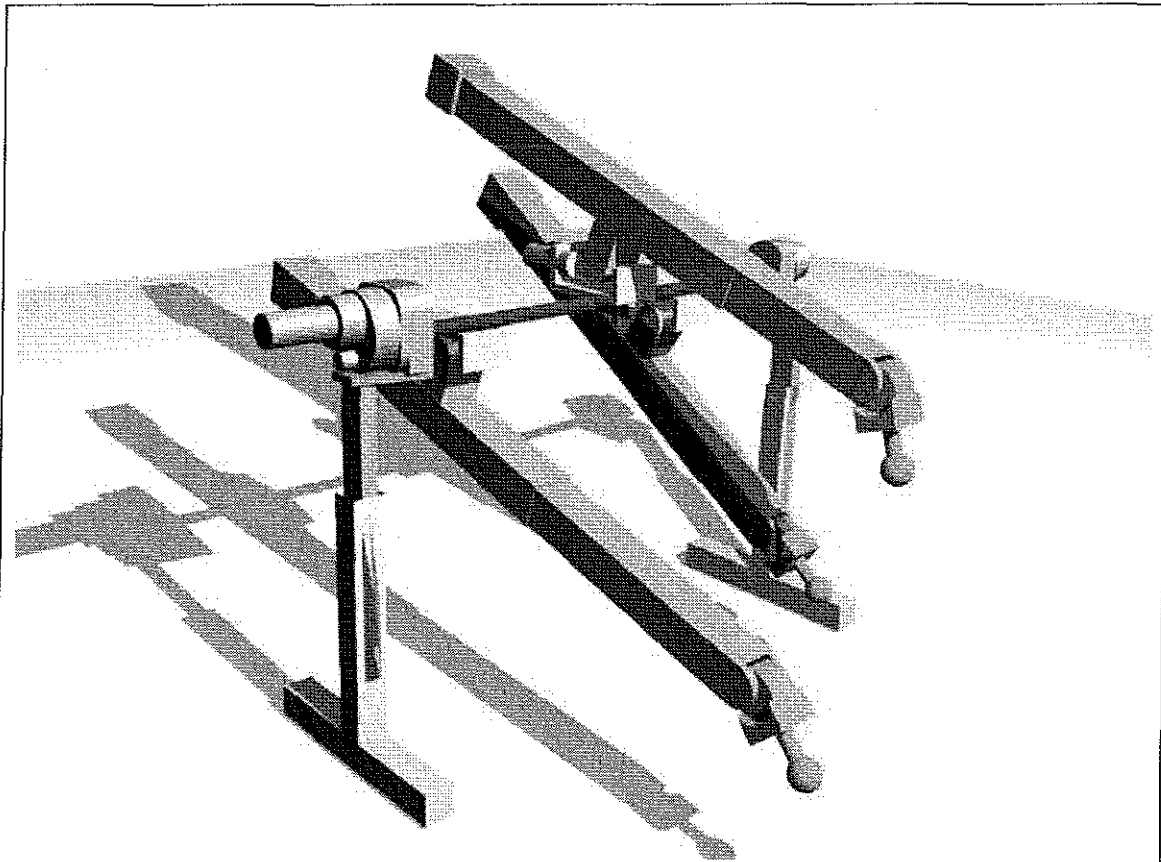


Figura 2.8: Model 3D del robot pòrtic de 3 braços

Un cop es té el disseny i les mides, aquest es modela i visualitza amb un software d'edició 3D. Aquest disseny virtual permet trobar, si hi ha, algun problema potencial ja que es pot fer una inspecció virtual amb diverses perspectives. Quan es dona el vist i plau a aquest models s'exporten en un format que permetin desenvolupar fàcilment el software del simulador.

El model ha de ser el més exacte possible, principalment les mides han de ser les que es van acordar en el disseny. També s'ha de tenir en compte fer un modelat per peces, de forma que el robot sigui un

conjunt de varies peces mòbils. El motiu principal és que el software de simulació haurà de moure les peces de forma independent.

S'ha de tenir present el nombre de triangles que componen el model. Un model molt detallat es veurà millor però tindrà més informació geomètrica a mostrar. Si es pretén fer un simulador que sigui interactiu és preferible fer un model amb menys informació geomètrica per tal que la visualització sigui un procés ràpid i es pugui dedicar la CPU a d'altres càlculs propis del procés.

Simulació i validació de l'arquitectura

Aquesta fase es porta a terme conjuntament amb cirurgians que són els qui realment jutgen la mobilitat del robot i la seva situació respecte el pacient.

Per portar a terme aquesta fase es desenvolupa un simulador 3D (figura 2.9) a partir dels models obtinguts a la fase anterior. L'estudi cinemàtic és una peça fonamental del simulador ja que proporciona les equacions per poder moure el robot de forma interactiva. En el cas del simulador només cal la cinemàtica inversa i la cinemàtica directa ja que l'estudi de la dinàmica i inèrcies requereix de dades, com el pes del robot i propietats dels motors de les quals encara no es disposa.

Tot i que al mercat hi ha un gran ventall de software que permeten simular sistemes robòtics s'ha preferit desenvolupar un de propi. El motiu principal és el fet de que els simuladors comercials són molt limitats a l'hora de permetre donar comandes al robot. Normalment la interacció amb aquest sistemes es fa amb teclat i ratolí i no permeten

ésser ampliat per a suportar un joystick o un sistema de posicionament magnètic.

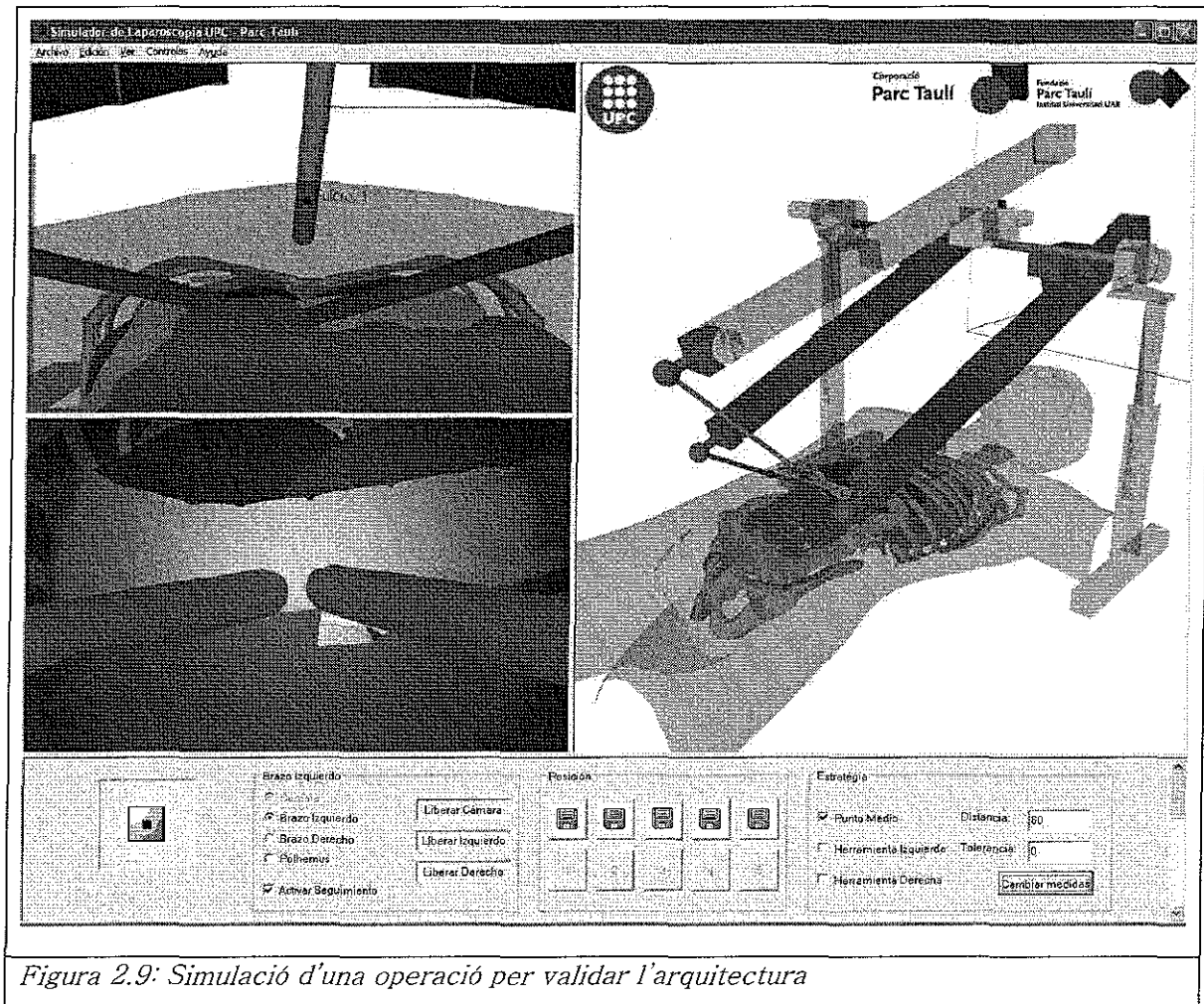


Figura 2.9: Simulació d'una operació per validar l'arquitectura

Un altre motiu és el fet de que són eines genèriques, per la qual cosa fan servir mètodes genèrics per implementar la cinemàtica. Els mètodes genèrics per implementar la cinemàtica inversa, com poden ser els mètodes basats en matrius jacobianes són imprecisos i computacionalment costosos.

Amb un simulador propi no només s'implementa la visualització i moviment del robot sinó que es pot provar, de forma virtual, el sistema de seguiment intel·ligent de les eines ja que es pot simular perfectament la vista de la càmera en tot moment.

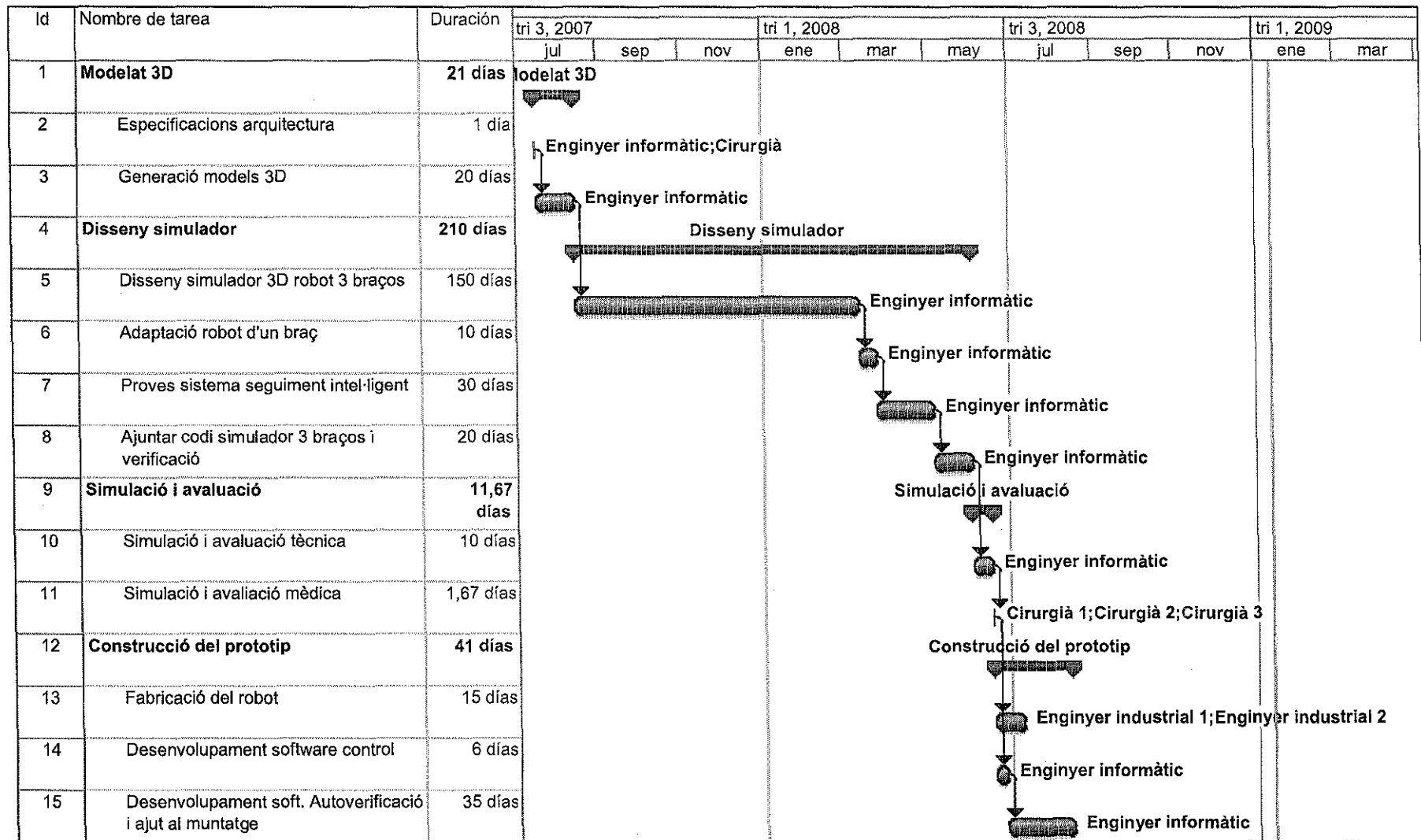
Amb aquest simulador i, conjuntament amb l'equip mèdic s'estudien diversos escenaris d'operacions i es valora la bondat de l'arquitectura simulant el procediment quirúrgic. Si es troba algun problema estructural es modifiquen els plànols i models i es prova una nova simulació fins que les mides i la mobilitat del robot sigui la que els cirurgians necessiten.

Disseny del Prototip

Un cop validada l'arquitectura del robot es fa el prototip amb les especificacions finals. De forma paral·lela es va desenvolupant el software de control del robot aprofitant part del software del simulador. A l'annex III es poden trobar els plànols referents al disseny del prototip.

Planificació

De cara a portar a terme el projecte s'ha fet una planificació en funció de les fases a seguir o dels objectius a complir a cada fase. El desenvolupament de tot el sistema té una durada prevista d'un any on intervenen 2 enginyers industrials, un enginyer informàtic i tres cirurgians.



Conclusions

En aquest capítol s'ha fet un breu recorregut de la història de la robòtica a la medicina, mostrant els avenços més importants. També s'ha mostrat el panorama actual i els diferents projectes i línies d'investigació que es porten a terme avui dia.

És en aquest marc actual en el que s'ha presentat un projecte conjunt entre el *departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial (ESAI)* i la *Fundació Parc Taulí* sobre nous sistemes assistencials per operacions per laparoscòpia. Una plataforma que farà més precisa i intuïtiva les tasques del metge.

També s'ha mostrat l'esquema general del projecte i quins són els sistemes implicats, indicant els principals i més prioritaris per l'estudi de la viabilitat del projecte.

Per acabar s'ha presentat la metodologia de treball, indicant com es du a terme la investigació sobre la viabilitat del projecte. Cal remarcar les etapes ben diferenciades i estructurades que afavoreixen l'agilitat d'aquest procés de recerca i aporten una garantia de viabilitat a l'hora d'invertir en el prototip.

Capítol 3:

Cinemàtica per al robot de 3 braços

Introducció

La cinemàtica és la ciència del moviment que tracta el tema sense tenir en compte les forces que el produeixen. Dintre d'aquesta ciència s'estudia la posició, velocitat i acceleració. Quan es parla de la cinemàtica d'un robot l'estudi implica obtenir les equacions que vinculen la posició final del manipulador amb els moviments de les articulacions.

Tenint en compte que l'estudi de velocitats i acceleracions d'un manipulador robòtic estan íntimament lligades a la construcció mecànica d'aquest (pes, inèrcies, etc...) només s'estudiaran els problemes relatius al posicionament del manipulador.

El primer problema a estudiar és el que es coneix comunament, al món de la robòtica com a problema de la **cinemàtica directa** (Direct Kinematics). En aquest cas es desitja saber on estarà l'extrem del manipulador quan el manipulador agafa una configuració determinada.

Hi ha molts paradigmes que donen resposta al problema de la DK, particularment s'estudiarà la resolució del manipulador amb el mètode de Denavit-Hartenberg.

També s'estudiarà el problema de la **cinemàtica inversa** (Inverse Kinematics) on el que es pretén es saber quina configuració han de prendre el manipulador per garantir que l'extrem es troba a una posició

desitjada. Es comentaran els principals mètodes que hi ha per resoldre-la i els problemes que, molt sovint, s'hi troben.

Cinemàtica directa (DK)

De la mateixa manera que els éssers vius tenen un sistema locomotor per moure's els robots es mouen gràcies a dos factors: els motors (que fan la feina dels músculs) i les articulacions. L'estudi de la posició del robot es basa en aquest últim. El mercat actual proveeix als fabricants d'un gran catàleg d'articulacions que permeten fer molts moviments diversos. Realment aquesta mobilitat s'aconsegueix combinant dues articulacions bàsiques: articulacions angulars (rotacions) i articulacions prismàtiques (translacions). És per això que gairebé tots els formalismes per modelitzar sistemes robòtics tenen en compte únicament aquest dos moviments possibles, combinant-los per obtenir la resta de moviments. Per exemple una articulació de cargol es pot expressar com a una rotació seguida d'una translació actuant totes dues sobre el mateix eix.

Suposant un robot amb n graus de llibertat $\sigma_i, i \in \{1, \dots, n\}$ és interessant saber on es trobarà l'extrem del robot en variar un d'aquest graus. Formalment la cinemàtica directa consisteix en resoldre una funció $F(\sigma_1, \dots, \sigma_n)$ tal que:

$$\{x, y, z, \alpha, \beta, \theta\} = F(\sigma_1, \dots, \sigma_n)$$

Amb aquesta funció i les seves derivades es pot estudiar com afecten petits canvis d'un grau de llibertat a la posició i orientació final, quelcom molt important en la programació d'un manipulador robòtic

seguint una trajectòria coneguda a fi de realitzar una tasca determinada, tal i com es pot veure a les cadenes de muntatge robotitzades.

Els formalismes matemàtics per modelitzar la cinemàtica directa són purament geomètrics. En destaquem els 4 principals:

- 1- Resolució mitjançant mètodes geomètrics directes.
- 2- Resolució mitjançant matrius homogènies.
- 3- Resolució mitjançant els paràmetres de Denavit-Hartenberg
- 4- Resolució mitjançant quaternions.

Resolució mitjançant mètodes geomètrics directes

Aquest és el mètode més trivial per solucionar la cinemàtica directa alhora de que no es un mètode genèric ja que consisteix en atacar el problema directament i trobar el conjunt d'equacions que el resolen. Només és recomanable per robots amb pocs graus de llibertat de manera que el seu desenvolupament geomètric sigui prou trivial de resoldre.

Resolució mitjançant matrius homogènies

Aquest mètode es basa en veure el robot com un conjunt de parts on cadascuna té un sistema de referència associat i definit respecte el sistema anterior. Per exemple, si sabem les coordenades del punt P respecte al sistema S_i ($P_{S(i)}$) i tenim la matriu de transformació del sistema S_i^{i-1} podem saber P respecte a S_{i-1} ($P_{S(i-1)}$) de la següent manera:

$$P_{S(i-1)} = S_i^{i-1} * P_{S(i)}$$

Les matrius homogènies que es fan servir són matrius de 4x4 on es codifica una rotació (submatriu R de 3x3) y una translació (T) de tal forma que la transformació següent:

$$P' = R * P + T$$

Es pot expressar com a:

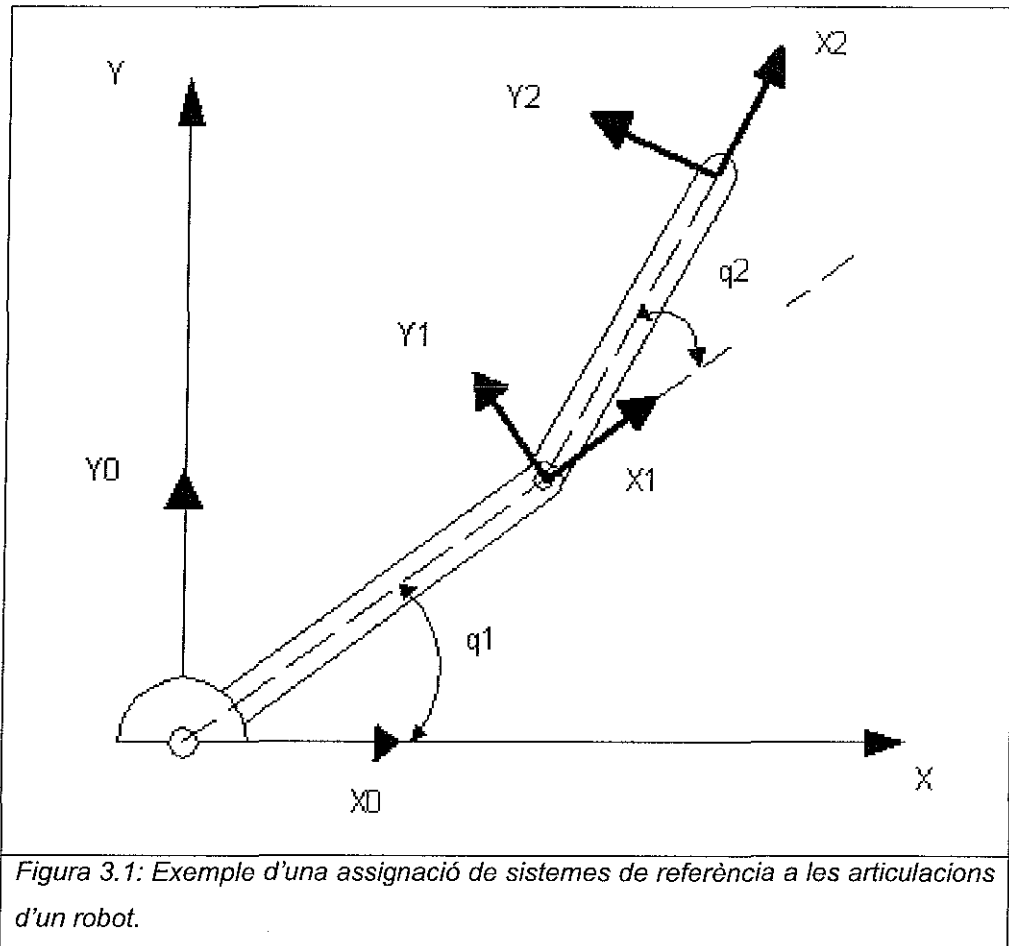
$$\begin{pmatrix} P' \\ 1 \end{pmatrix} = \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix} * \begin{pmatrix} P \\ 1 \end{pmatrix}$$

Hi ha varies formes per expressar les rotacions, essent la més comuna les rotacions amb angles d'Èuler:

$$R = R(z, \alpha) * R(y, \beta) * R(z, \theta)$$

El robot es divideix en tantes parts com articulacions tingui. A cada part se li associa un sistema de coordenades local amb origen el centre de l'articulació. Cada sistema de coordenades es defineix respecte al sistema anterior. Finalment es defineix un sistema de coordenades respecte l'ultima articulació per a marcar l'extrem del braç, on hi anirà una eina.

Per acabar, i tenint en compte que cada sistema de coordenades es defineix respecte de l'anterior fa falta afegir el sistema de coordenades de la base del robot que servirà per indicar la posició d'aquest respecte al sistema de referència global(figura 3.1).



D'aquesta forma, si es vol saber la matriu de E de l'element terminal es demostra que:

$$E = S_n^0 = S_1^0 * \dots * S_n^{n-1}$$

On s'obté que:

$$E = \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix}$$

On R conté, per columnes els vectors de la base S_n^0 i $T = O_{S(n)} - O_{S(0)}$ del que es dedueix que T és el punt extrem del manipulador expressat en la base S_0 .

Resolució mitjançant els paràmetres de Denavit-Hartenberg

Aquesta es la convenció més usada per la resolució de manipuladors lineals, on la posició d'una articulació depèn únicament de la posició de l'articulació anterior. És un cas particular de matrius homogènies on cada matriu queda determinat per 4 paràmetres: a , α , θ , d'on 3 d'ells són intrínsecs al robot i el 4rt és el grau de llibertat i depèn de si l'articulació és angular (θ) o prismàtica (d).

A més l'algorisme diu com s'han de construir els sistemes de referència i obtenir aquests paràmetres de forma sistemàtica. De fet, aquest paràmetres s'estableixen entre dos sistemes de coordenades consecutius. La millora que té aquesta convenció es el fet de que una matriu queda determinada per 4 paràmetres en comptes de 6 (3 per la posició i 3 per l'orientació).

S'ha de dir que la bibliografia que tracta sobre l'algorisme de Denavit-Hartenberg sol fer servir convencions diferents, sobretot a l'hora de compondre la matriu. De cara a aquest projecte s'ha optat per la convenció exposada a *"Introduction to Robotics"* de John J. Craig per ser bibliografia bàsica i indispensable en l'estudi de la robòtica.

El procediment per construir els sistemes de referència és el següent:

- 1- Identificar els eixos de les articulacions i dibuixar línies infinites sobre ells, per als passos del 2 al 5 considerar dues d'aquestes línies adjacents (en els eixos i i $i+1$).

- 2- Identificar la perpendicular comuna entre ells, o el punt comú d'intersecció. En el punt d'intersecció, o en el punt en que la perpendicular comuna es troba amb l' i -èssim eix, assignar l'origen del sistema associat a la peça.
- 3- Assignar l'eix Z_i per que apunti sobre l' i -èssim eix d'articulació.
- 4- Assignar l'eix X_i per que sigui normal al pla que conté els dos eixos.
- 5- Assignar l'eix Y_i per que el sistema de referència sigui dextrogir. Aquest es pot aconseguir mitjançant el producte vectorial de Z_i i X_i :

$$Y_i = Z_i \times X_i$$

- 6- Assignar S_0 per tal que concordi amb S_1 quan la primera variable d'articulació sigui 0. Per N seleccionar la ubicació de l'origen i la direcció d' X_N lliurement, però generalment de forma que faci que la major part dels paràmetres dels vincles sigui 0.

Un cop assignats els eixos, per un sistema S_i^{i-1} els paràmetres es defineixen com:

a_i = distància de Z_i a Z_{i+1} mesurada sobre X_i .

α_i = angle de Z_i a Z_{i+1} mesurat sobre X_i .

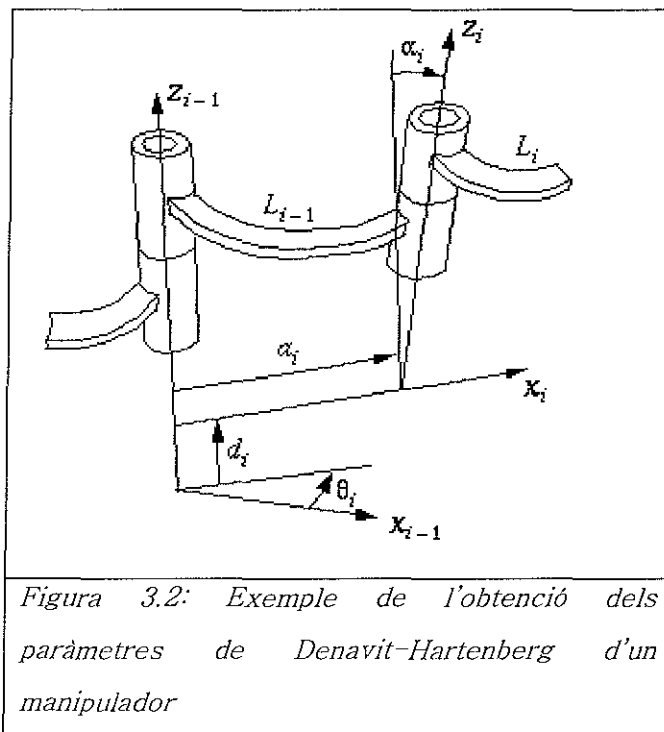
d_i = distància de X_{i-1} a X_i mesurada sobre Z_i .

θ_i = angle de X_{i-1} a X_i mesurat sobre Z_i .

Amb aquest 4 paràmetres una matriu homogènia S_i^{i-1} s'especifica com:

$$S_i^{i-1} = R(X, \alpha_{i-1}) * T(X, a_{i-1}) * R(Z, \theta_i) * T(Z, d_i)$$

Es pot veure un exemple d'aplicació d'aquest mètode a la figura 3.2.



Resolució mitjançant quaternions

No s'aprofundirà sobre l'estudi dels quaternions pel fet de que la seva àlgebra i comprensió són molt complexos i per entrar-ne en detall fan falta coneixements superiors en matemàtiques. Només esmentar que les matrius de transformacions homogènies i els quaternions són mètodes alternatius per representar transformacions de rotació i translació. En

conclusió: és possible fer servir un o l'altre per resoldre el problema cinemàtic directe d'un robot.

Resolució de la cinemàtica directa del manipulador de 3 braços

Seguint l'àmplia bibliografia al respecte, s'ha optat pel mètode de Denavit-Hartenberg per resoldre la cinemàtica del robot d'estudi. S'ha d'esmentar que aquest algorisme està pensat per sistemes amb un únic manipulador i aquest robot en té 3. És per això que es va optar en veure aquest robot com a 3 robots independents i tractar per software el fet de compartir una articulació.

Un cop esmentada aquesta simplificació s'ha de veure que els braços inferiors tenen la mateixa configuració que el braç superior (figura 3.3) però invertida col·loquen invertits (figura 3.4), amb la qual cosa s'han de fer dues taules de Denavit-Hartenberg. Els termes X_i fan referència als graus de llibertat i els termes Y_i fan referència als paràmetres ajustables per ergonomia.

Els paràmetres del primer braç són:

Tipus	α	a	θ	d
Prismàtica	0.0	0.0	0.0	Y_1
Prismàtica	$\pi/2$	0.0	$\pi/2$	X_2
Angular	0.0	20.0	X_3	0.0
Prismàtica	$\pi/2$	90.0	0.0	Y_4
Prismàtica	0.0	0.0	0.0	X_5
Angular	$-\pi/2$	0.0	Y_6	0.0

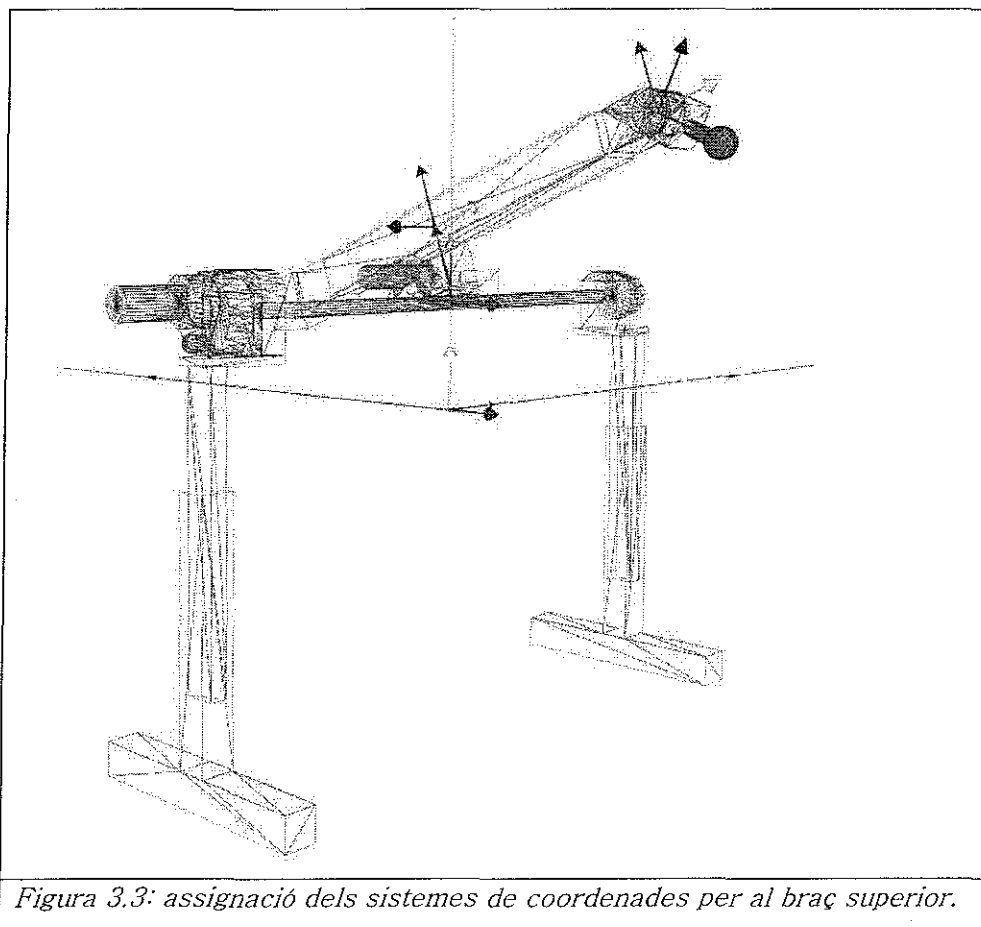


Figura 3.3: assignació dels sistemes de coordenades per al braç superior.

Per als braços inferiors els paràmetres són:

Tipus:	α	a	θ	d
Prismàtica	0.0	0.0	0.0	Y_1
Prismàtica	$\pi/2$	0.0	$\pi/2$	X_2
Angular	0.0	20.0	X_3	0.0
Prismàtica	$\pi/2$	90.0	π	Y_4
Prismàtica	0.0	0.0	0.0	X_5
Angular	$-\pi/2$	0.0	Y_6	0.0

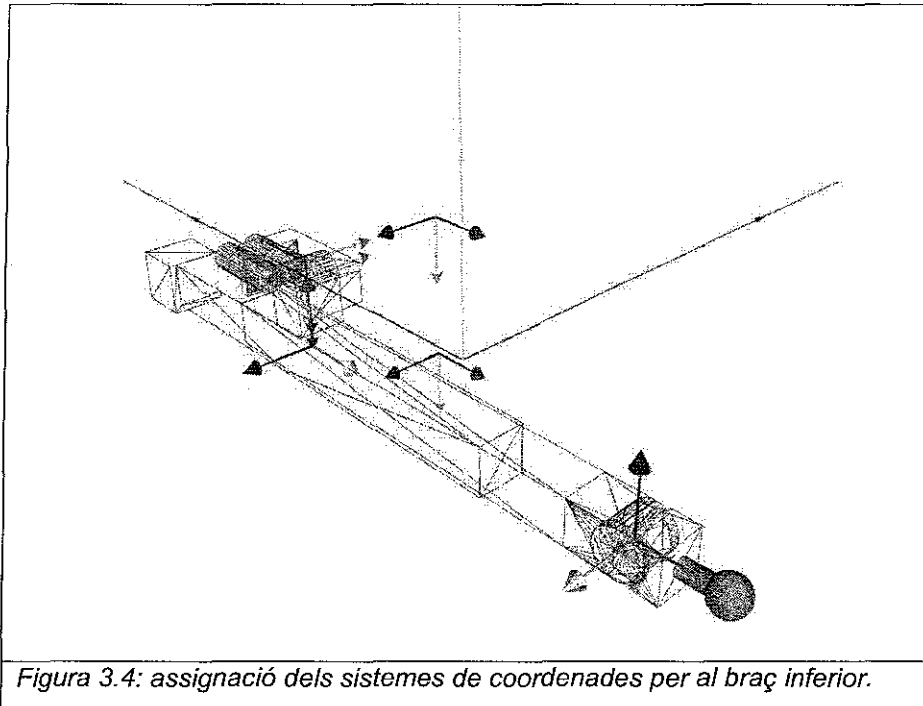


Figura 3.4: assignació dels sistemes de coordenades per al braç inferior.

Com es pot apreciar l'única diferència entre la taula del braç superior i la dels inferiors es θ_4 que a la taula 1 és 0 i a la taula 2 és π . Tot i això fan falta més canvis per invertir els dos braços inferiors: el sistema S_0 és diferent per al braç superior i els inferiors.

Per al braç superior:

$$S_0 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ Rot(z, -\pi/2) & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

I per als inferiors:

$$S_0 = \begin{pmatrix} 0 & 50 & -10 & 1 \\ Rot(z, -\pi/2) * Rot(x, \pi) & 0 & 0 & 0 \end{pmatrix}$$

Cinemàtica inversa (IK)

Amb la resolució de la cinemàtica directa s'ha vist com moure el robot però el problema interessant és saber com moure'l cap a un lloc determinat . Això és, a grans trets el què es resol amb la cinemàtica inversa. Si la cinemàtica directa es pot descriure com una funció $F(\sigma_1, \dots, \sigma_n)$ tal que:

$$\{x, y, z, \alpha, \beta, \theta\} = F(\sigma_1, \dots, \sigma_n)$$

La cinemàtica inversa és la funció $F^{-1}(x, y, z)$ tal que:

$$(\sigma_1, \dots, \sigma_n) = F^{-1}(x, y, z)$$

Més intuïtivament: la cinemàtica inversa obté els valors de les articulacions $\sigma_1, \dots, \sigma_n$ per aconseguir que l'extrem del manipular sigui al punt $P = (x, y, z)$.

El problema principal de la cinemàtica inversa recau en què, al ser $F(\sigma_1, \dots, \sigma_n)$ una funció suprajectiva es dedueix que la funció $F^{-1}(x, y, z)$ no serà injectiva. Dit d'una altre forma: la solució de la cinemàtica inversa pot no ser única i s'han d'establir paràmetres a fi d'escollir la millor. Un altre problema ve del fet que encara no hi ha mètodes tancats per trobar una solució general prou precisa i, en molts casos s'ha de treballar amb solucions particulars.

Una particularització del problema permet reduir el nombre de solucions. Es tracta d'imposar que la solució sigui un sistema de coordenades concret en comptes d'un punt de destí:

$$(\sigma_1, \dots, \sigma_n) = F^{-1}(x, y, z, \alpha, \beta, \theta)$$

A la pràctica aquesta solució implica que es vol col·locar l'element terminal del manipulador en una posició i orientació determinades. Un problema molt comú quan els robots estan pensats per a portar una eina. Com que s'estan imposant 3 condicions finals addicionals es redueix bastant el nombre de solucions possibles.

Si els mètodes per resoldre la cinemàtica directa eren purament geomètrics a la cinemàtica inversa es sol recórrer, a més, a mètodes numèrics o inclús a tècniques d'intel·ligència artificial per triar una bona solució, sobre tot si l'arquitectura del robot té graus de llibertat redundants.

Els mètodes més comuns son:

1. Resolució mitjançant mètodes geomètrics directes
2. Resolució a partir de la matriu de transformació homogènia
3. Resolució mitjançant mètodes numèrics
4. Resolució mitjançant la matriu jacobiana

Tots ells tenen punts forts i febles que els fan adients per uns certs casos. A continuació s'explicaran més detalladament.

Resolució mitjançant mètodes geomètrics directes

Idènticament a com es va explicar amb la cinemàtica directa, aquest mètode consisteix en plantejar, per un robot concret el sistema d'equacions en funció dels valors de les articulacions per arribar al punt

final. Aquest mètode comporta greus inconvenients: múltiples solucions poden generar equacions d'ordre superior a 2. A més, el fet de treballar amb angles fa que a les equacions es vegin implicades funcions trigonomètriques, a vegades difícils de tractar per la seva dependència.

Com a punts forts destaquen la rapidesa amb la que es pot computar la cinemàtica inversa i la seva precisió. Això fa d'aquest mètode el més adient per aplicacions interactives i de precisió. Aquesta solució és aplicable i recomanable quan l'arquitectura del robot es prou senzilla i no hi ha graus de llibertat redundants que augmentin el nombre de solucions possibles o faci difícil extreure les solucions del sistema.

Resolució a partir de la matriu de transformació homogènia

Aquest mètode és una forma d'aconseguir un sistema d'equacions les quals no es veuen tan “a ull” i requereix el càlcul de la cinemàtica directa mitjançant matrius homogènies (o Denavit-Hartenberg com a cas particular). És un mètode que consisteix en extreure un sistema d'equacions a partir de la manipulació algebraica i igualació d'expressions matricials implicades. Un exemple podria ser considerar un robot amb 4 matrius implicades de tal forma que

$$T = A_1^0 * A_2^1 * A_3^2 * A_4^3$$

Es fàcil extreure dues relacions:

$$(A_1^0)^{-1} * T = A_2^1 * A_3^2$$

$$(A_2^1)^{-1} * (A_1^0)^{-1} * T = A_3^2$$

Si T és el sistema de referència de destí obtenim que:

$$T = \begin{pmatrix} B & P \\ 0 & 1 \end{pmatrix}$$

On $B = \langle e_1, e_2, e_3 \rangle$ són els vectors directors del sistema de referència T i $P = (x, y, z)$ el punt origen del T expressats en el sistema de coordenades global. Per resoldre el problema es pot agafar la primera igualtat i aplicar la inversió de matrius de transformacions homogènies.

$$M = \begin{pmatrix} B & P \\ 0 & 1 \end{pmatrix} \Leftrightarrow M^{-1} = \begin{pmatrix} B' & -B' * P \\ 0 & 1 \end{pmatrix}$$

Siguin:

$$X = (A_1^0)^{-1} * T$$

$$Y = A_2^1 * A_3^2$$

S'obté que

$$X = Y \Leftrightarrow x_{i,j} = y_{i,j}$$

Amb aquesta igualtat de matrius s'obtenen fins a 16 equacions, de les quals s'haurien d'escollir els 4 (graus de llibertat) més adients.

Aquest mètode té els mateixos avantatges i inconvenients que la resolució per mètodes geomètrics directes ja que aquí l'objectiu és trobar les equacions que computarà el software però seguint una certa metodologia. L'ús d'aquest mètode es du a terme en temps de disseny, amb la qual cosa no implica un major cost computacional que els mètodes geomètrics directes.

Resolució mitjançant mètodes numèrics

Aquests mètodes fan servir algorismes de prova i error per trobar una solució aproximada al punt demanat. Hi ha moltes opcions depenent del grau de complexitat del robot. La versió més senzilla d'aquest mètode seria una cerca dicotòmica de tots els graus de llibertat implicats dintre del seus valors mínims i màxims o un *backtracking*. Versions més acurades incorporen un conjunt de solucions pre calculades i fan cerca dintre d'aquest subespai partint de la solució més propera i refinant-la. Aquest pre càlcul permet fer servir algorismes de cerca local com *Hill Climbing* o *Simulated Annealing*. Per sistemes robòtics complexos s'arriben a fer servir xarxes neuronals amb una bona fase d'aprenentatge.

Aquest mètodes serveixen per trobar solucions aproximades i la seva complexitat (depenent en gran part de la precisió desitjada) no els fan adequats per software interactiu. S'ha de tenir en compte que a cada iteració es comprovarà la solució mitjançant la cinemàtica directa a fi d'encaminar la següent iteració. És per això que el cost mig d'aquest mètode serà

$$C(k,n) = k * O(n * 4^3)$$

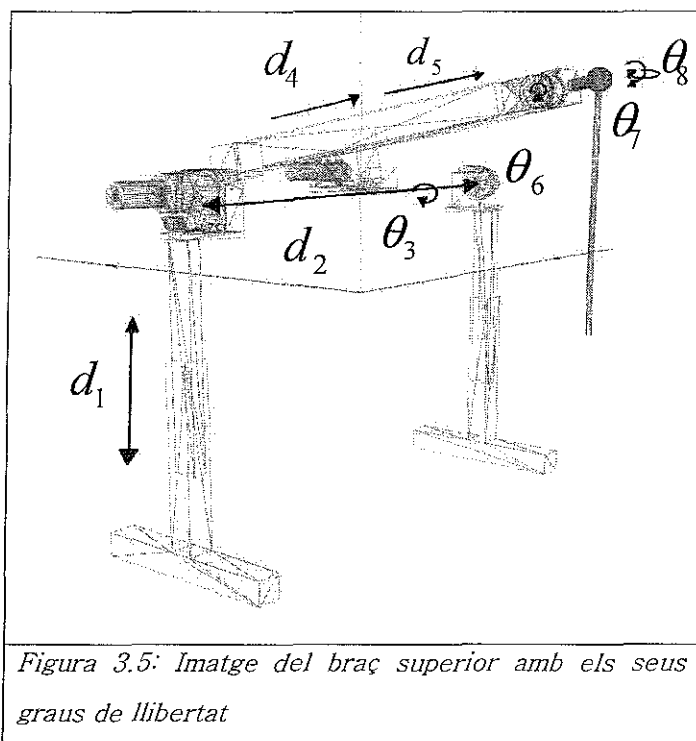
On k és el nombre d'iteracions i n és el nombre de graus de llibertat. Ja que les matrius són de 4x4 el cost del càlcul de la cinemàtica directa és de $O(n * 4^3)$.

Es recomana emprar aquest mètodes en arquitectures amb moltes articulacions i graus de llibertat redundants. Els mètodes basats en

tècniques d'IA (*Hill Climbing*) permeten establir criteris de “millor solució” a fi de que l'algorisme convergeixi en un estat òptim. Les tècniques més sofisticades basades en aprenentatge o en cerca local tenen com a objectiu reduir la k del cost de l'algorisme.

Resolució de la cinemàtica inversa del manipulador de 3 braços

Igualment a com es va fer amb la cinemàtica directa és millor tractar els braços de forma independent. De fet tots 3 braços tenen la mateixa construcció i es diferencien en els sistemes de coordenades.



El braç robot té 3 graus de llibertat (d_2, θ_3, d_5), i 3 paràmetres manualment ajustables (d_1, d_4, θ_6) i dos graus de llibertat passius (θ_7, θ_8) en el punt de subjecció amb l'eina (figura 3.5). El problema que s'ha de resoldre és trobar quins valors han de rebre els 5 graus de llibertat (tant actius com passius) de manera que l'extrem de l'eina del robot sigui al

punt P i amb una orientació donada pels angles α i β . La funció de cinemàtica inversa seria de la forma:

$$(d_2, \theta_3, d_5, \theta_7, \theta_8) = F(P, \alpha, \beta, d_1, d_4, \theta_6)$$

Ja que els graus θ_7 i θ_8 són passius es pot obtenir la cinemàtica inversa del robot sense tenir-los en compte si considerem resoldre la cinemàtica per al punt d'unió entre el robot i l'eina P' en comptes de l'extrem de l'eina. Si VP és el vector que va de P a P' quan α i β són 0 tenim que:

$$P' = \begin{pmatrix} Rot(x, \alpha) * Rot(y, \beta) & VP \\ 0 & 1 \end{pmatrix} * P$$

I la funció a resoldre quedaria com:

$$(d_2, \theta_3, d_5) = F(P', d_1, d_4, \theta_6)$$

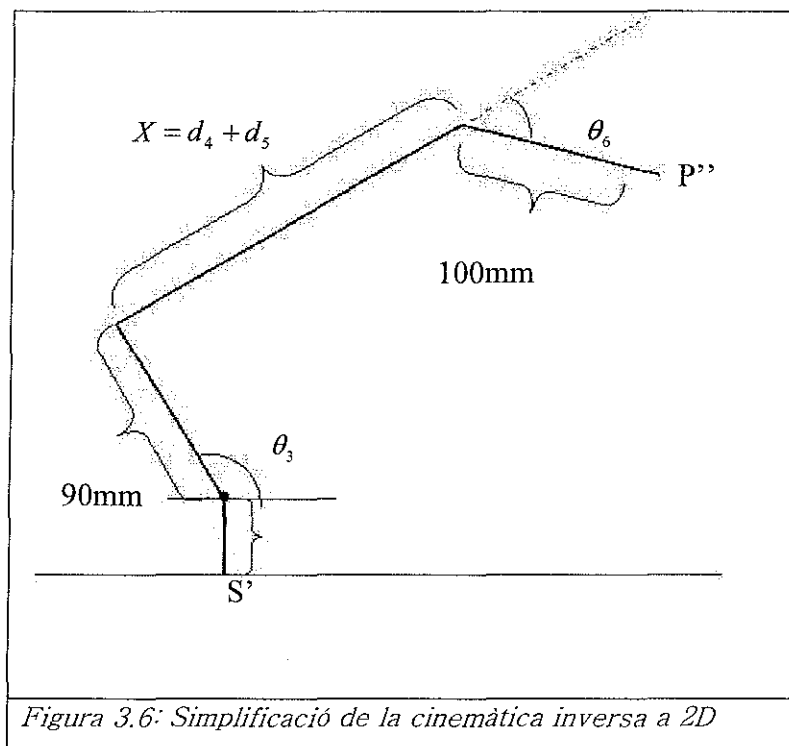
Primer s'ha d'ajustar P' ja que es troba expressat en el sistema de referència global i interessa que estigui expressat en funció del robot. De fet interessa que estigui expressat respecte S' , on

$$S' = S_3^0 = S_1^0 * S_2^1 * S_3^2$$

i els S_j^i són els sistemes de coordenades extrets mitjançant Denavit-Hartenberg (i ja tenen en compte el valor d_1). De tal forma que:

$$P'' = S'^{-1} * P'$$

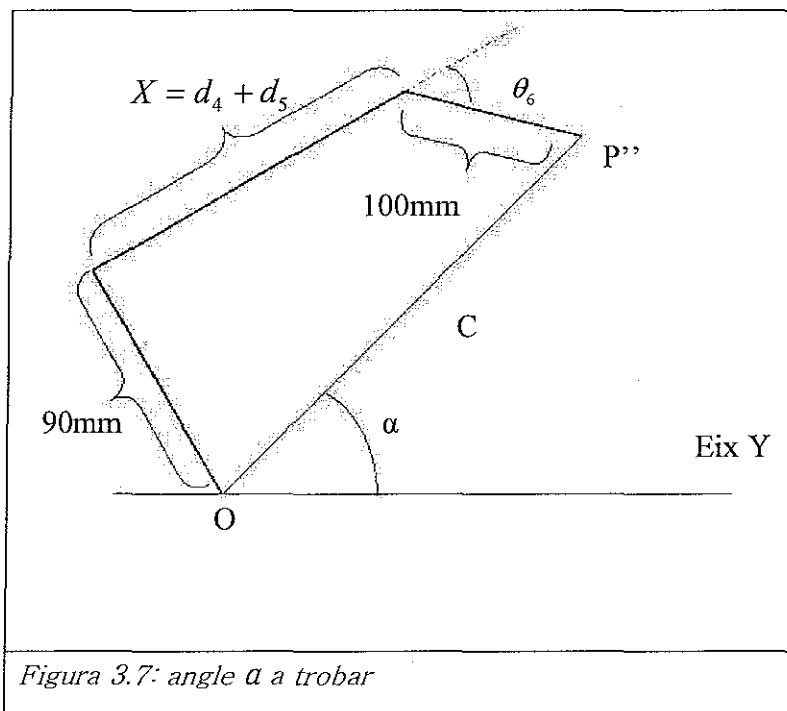
L'obtenció del punt P'' en funció de la posició del robot és el que fa que el conjunt d'equacions resultants sigui vàlid per tots 3 braços. S'ha de tenir en compte que el valor de d_2 que afecta a S_3^2 ha de ser 0 en el moment de calcular S' . Amb aquesta simplificació es troba que $d_2 = P''x$. La resta de la resolució es fa sobre $R^2(y,z)$ en comptes de $R^3(x,y,z)$ (figura 3.6).



A continuació s'ha de mirar l'angle α que forma el vector OP'' amb l'eix Y (figura 3.7).

Aquest angle s'obté com a:

$$\alpha = \arccos \left(\frac{OP'' \cdot Y}{|OP''| \cdot |Y|} \right)$$



Però també es necessita el signe, aquest s'aconsegueix mirant quin és el signe del determinant que formen la matriu dels vectors OP'' , l'eix Y i l'eix X . (es fa servir l'eix X perquè és un vector perpendicular als altres dos).

$$\text{signe} = \text{signe} \begin{vmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{vmatrix}$$

La construcció del segments del robot amb la recta OP'' genera una figura trapezoïdal. Amb l'ajut de l'angle θ_6 es pot aproximar aquesta figura a un rectangle, més còmode de resoldre (figura 3.8).

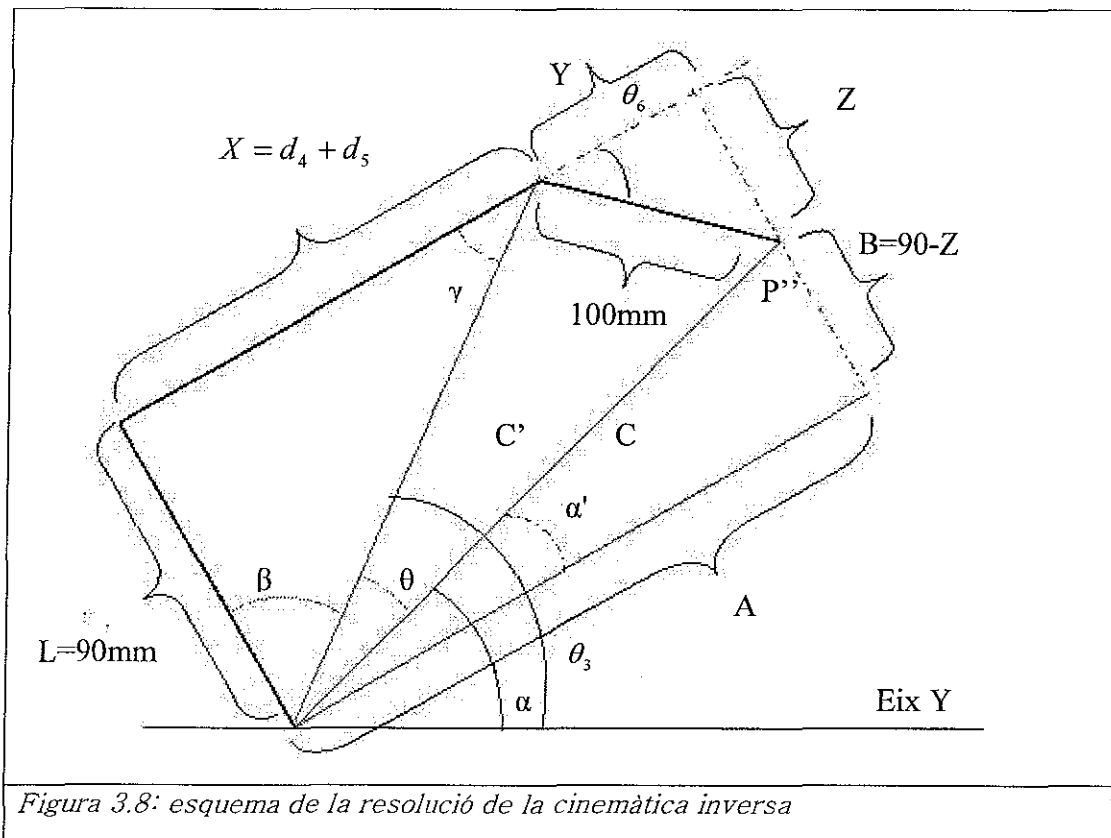


Figura 3.8: esquema de la resolució de la cinemàtica inversa

Les relacions simples:

$$A = X + Y$$

$$C = OP''$$

$$Y = 100 * \cos(\theta_6)$$

$$Z = 100 * \sin(\theta_6)$$

Amb B i C és resol A de la següent manera:

$$\alpha' = \arcsin(B/C)$$

$$A = C * \cos(\alpha')$$

Un cop resolt A es passa a obtenir d_5 .

$$X = A - Y$$

$$d_5 = X - d_4$$

Finalment, per obtenir θ_3 es fa servir el triangle XLC' i el fet de que L i A formen un angle de 90° .

$$\gamma = \arctan(L/X)$$

$$\beta = \pi/2 - \gamma$$

$$\theta = \pi/2 - (\alpha' + \beta)$$

$$\theta_3 = \alpha + \theta$$

Conclusions

En aquest capítol s'ha mostrat un subconjunt de les tècniques actuals a l'hora de fer els estudis cinemàtics d'un manipulador, principalment les diverses estratègies tant per la cinemàtica directa com per la inversa. D'aquestes tècniques s'han estudiat els seus pros i contres a fi de veure quines són les més adients per uns escenaris concrets.

L'estudi de la cinemàtica d'un manipulador és una part molt important a l'hora de dissenyar un robot, ja que és aquest qui valida o no la intenció amb que es va dissenyar i la seva viabilitat. A més, aquest estudi permet obtenir les equacions per moure i controlar el manipulador. Sense aquestes equacions el robot dissenyat mai podrà complir amb la seva finalitat.

Capítol 4:

Software de simulació interactiva per al robot pòrtic de 3 braços

Introducció

La simulació 3D i amb realitat virtual és una eina molt potent que s'ha introduït de forma massiva als processos de disseny de molta maquinaria actual. Aquesta permet estudiar diversos aspectes d'un producte en la fase de disseny sense haver de fer un prototip amb l'elevat cost que comporta.

A més la simulació d'un disseny pot proporcionar un *feed-back* en una etapa molt inicial del procés de disseny, fent que els errors es puguin detectar i ser corregits aviat i fent del disseny una etapa interactiva que ajuda al refinament de la solució a fi de que es pugui donar una certa garantia sobre la validesa del producte final.

Aquest conjunt d'avantatges és el que va impulsar el desenvolupament d'un simulador interactiu del robot pòrtic. El simulador es va emprar per diverses tasques:

1. Estudi i ajusts de les dimensions per treballar dintre dels paràmetres de mides segons estudis antropomètrics.
2. Estudi i validació de les equacions de la cinemàtica tant directa com inversa.
3. Incorporació d'ajusts ergonòmics.

4. Estudi de col·lisions en diverses simulacions d'operacions
5. Estudi del sistema de *tracking* de les eines.

Amb aquest simulador es van poder validar molts aspectes i tècniques sense haver d'esperar a tenir el prototip. També es varen detectar problemes no deduïbles directament sobre plànols i que únicament es van trobar quan es va simular el moviment del robot.

Arquitectura: diagrames de classes i sistemes implicats

El simulador (figura 4.1) es va pensar per què fos fàcilment extensible i adaptable a d'altres robots/tasques. Es va dissenyar amb una filosofia clarament orientada a objectes on aquests formaven part d'un sistema específic (figura 4.2).

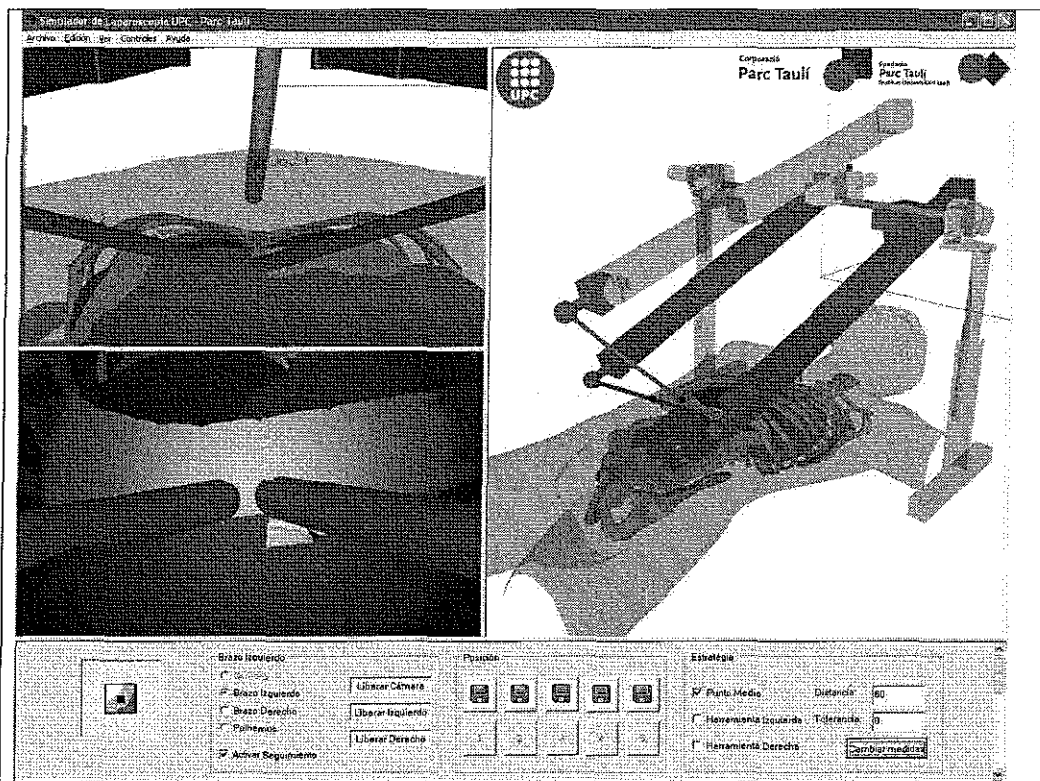


Figura 4.1: Simulador del robot pòrtic de 3 braços

Dintre el simulador hi ha cinc sistemes ben diferenciats i un controlador

1. Sistema 3D
2. Sistema de representació
3. Sistema RPQ
4. Sistema d'E/S
5. Sistema sèrie

Seguint les filosofies de diversos motors de *render* es va separar la part de geometria i representació del context de visionat, és per això que hi ha un sistema per configurar les àrees de dibuix i un sistema de representació on es prepara la configuració geomètrica a dibuixar.

El sistema RPQ és un conjunt de llibreries destinat a implementar la cinemàtica directa resolta mitjançant l'algorisme de Denavit-Hartenberg i la detecció de col·lisions.

El sistema d'E/S permet implementar la interacció amb diversos dispositius d'entrada/sortida. La versió actual inclou control amb joystick compatible amb *force-feedback* de tal forma que si l'usuari mou el robot cap a una posició no vàlida o hi ha una col·lisió entre els braços el joystick tremola.

El sistema sèrie va ser un afegit posterior destinat a comunicar una adaptació de l'aplicació amb un robot real mitjançant un port sèrie i un protocol que es va especificar entre el software i la controladora del robot.

El controlador té com a finalitat comunicar els diferents sistemes entre ells i amb la interfície d'usuari. També s'implementen certes estratègies i algorismes com són la resolució de la cinemàtica inversa i el seguiment de les eines per part del braç de la càmera (tracking).

Sistema 3D

Aquest sistema és fàcilment identificable en l'aplicació ja que són les tres regions sobre les que es dibuixa la geometria. Està implementada amb *OpenGL* i configura un context de dibuixat, les seves propietats (culling, z-buffer, ús de transparències, etc) i el seu comportament conforme a events de teclat (zooms, rotacions, translacions, etc).

L'elecció d'*OpenGL* com a sistema gràfic va ser en base a la seva orientació purament matemàtica i la seva filosofia de treball amb sistemes de coordenades que s'adequava prou bé a l'algorisme de Denavit-Hartenberg. *OpenGL* és portable i està implementat als principals sistemes operatius a més de ser un estàndard implementat per hardware a la majoria de targetes gràfiques del mercat.

El simulador mostra tres perspectives diferents de l'escena. Hi ha una perspectiva general on l'usuari pot desplaçar i rotar la càmera per situar-la allà on més li convingui. Hi ha una àrea on es mostra la perspectiva des de la càmera laparoscòpica, de forma que es té una versió simulada de la vista del metge a l'hora d'operar. Una tercera perspectiva mostra l'eina quirúrgica des d'un tall axial per donar una idea més exacta del lloc on s'està treballant.

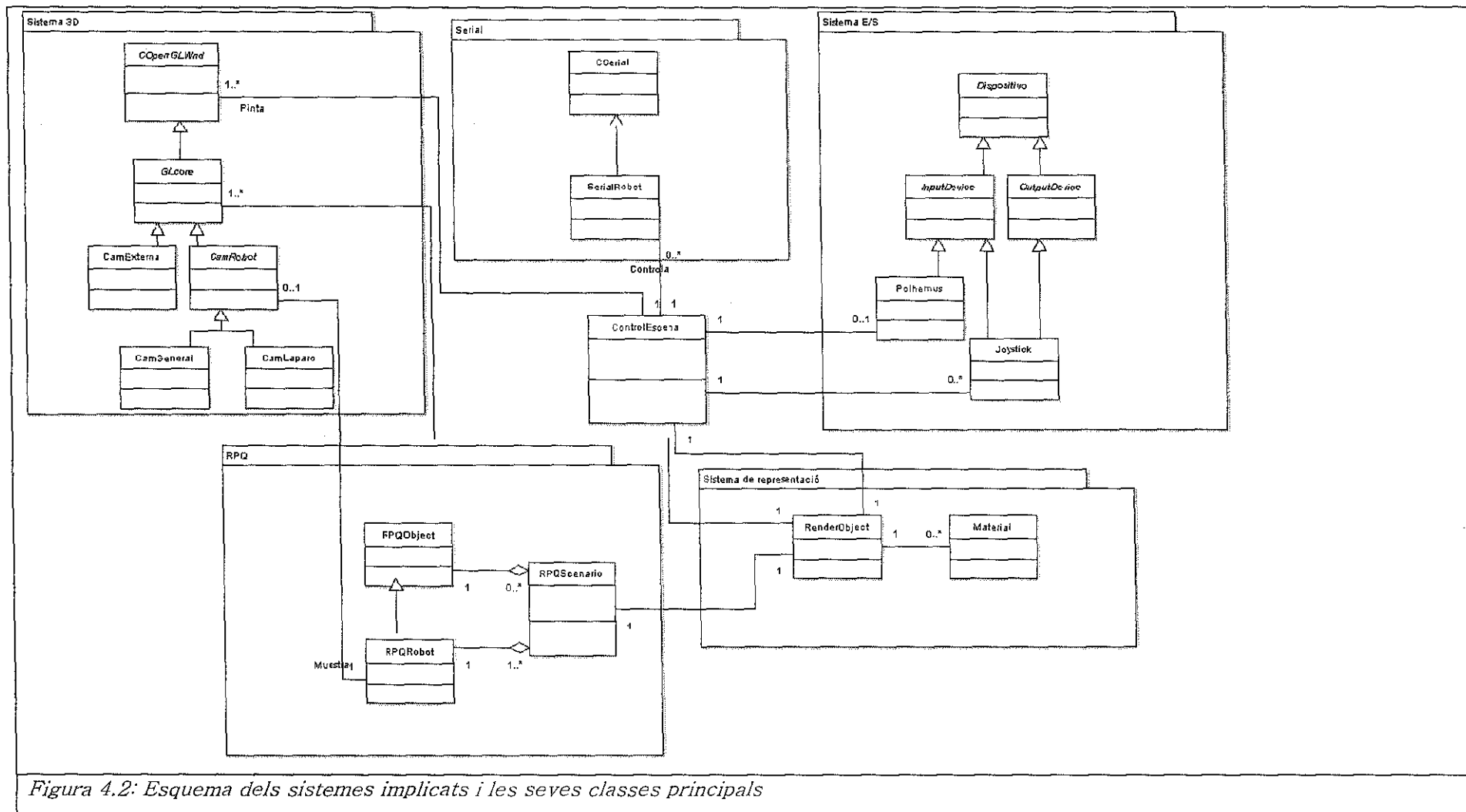


Figura 4.2: Esquema dels sistemes implicats i les seves classes principals

Sistema de representació

Tota la informació geomètrica mostrada dels models RPQ, materials, etc... són responsabilitat d'aquest sistema. Entre d'altres tasques cal mencionar la inicialització i generació de les *Display Lists*, una tècnica emprada per *OpenGL* per la qual un conjunt de primitives gràfiques queden preparades a la tarja gràfica per ser mostrades posteriorment. Aquest mètode fa que el dibuixat de primitives carregades en una *Display List* sigui fins a tres cops més ràpid que un dibuixat normal.

La separació entre el context de dibuixat i la informació a dibuixar permet estendre i adaptar millor el software a futures necessitats. Aquesta es la filosofia amb que treballen una bona part dels motors de renders actuals. Normalment el context de dibuixat acostuma a estar lligat a una plataforma/S.O. específics i és un element software difícil de portar a d'altres sistemes.

El sistema de representació també gestiona els materials. Visualment és el comportament respecte de la llum que té cada objecte a dibuixar. El seu control permet canviar parts del robot de color per indicar col·lisions, part seleccionada, etc.

En aquest punt també hi intervé informació adquirida en el modelat com són els vectors normals a cada vèrtex del model. Això és útil per donar una millor qualitat a la visualització. Fent servir les normals per vèrtex s'aconsegueix un efecte de suavitzat dels models sense haver d'augmentar el nombre de polígons del model.

Sistema RPQ

RPQ (Robot Proximity Queries) és una llibreria propietat del departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial (ESAI) destinada a implementar les estructures de dades i algorismes relacionats amb la posició i moviment dels components del robot. També s'encarrega de la detecció de col·lisions (figura 4.3). RPQ conté un ampli ventall de funcions per manipular matrius de transformacions homogènies i implementa l'algorisme de Denavit-Hartenberg.

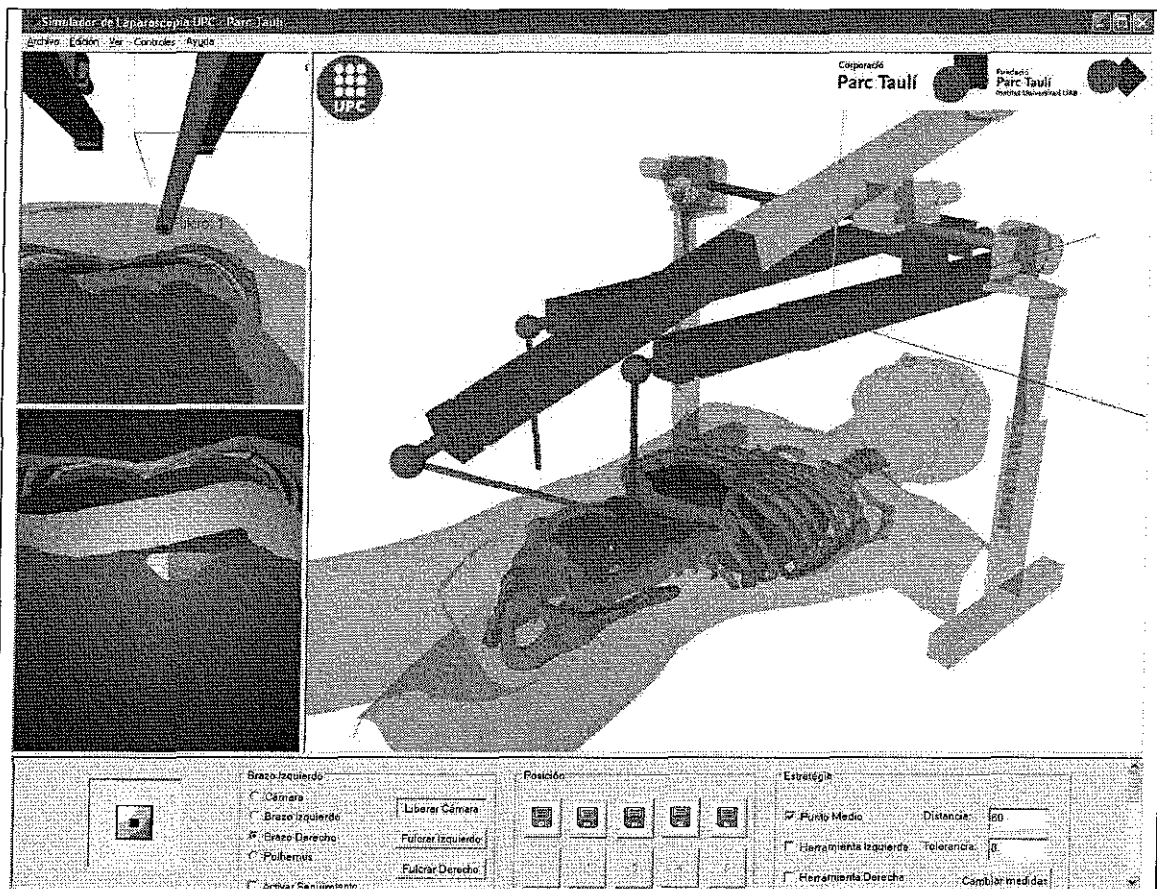
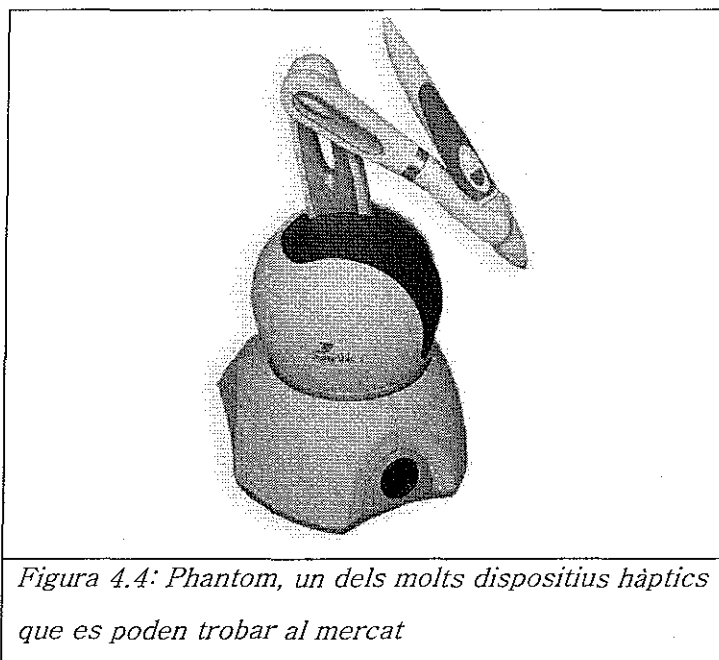


Figura 4.3: Detecció d'una col·lisió entre dos robots. En vermell es mostren els elements implicats en la col·lisió

Hi ha moltes llibreries per modelitzar sistemes robòtics basats en l'algorisme de Denavit-Hartenberg. La principal motivació que va tenir del *departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial* per implementar la seva pròpia llibreria és que cap de les existents incorporava un sistema de detecció de distàncies i col·lisions. RPQ implementa les col·lisions mitjançant PQP (*Proximity Query Package*: <http://www.cs.unc.edu/~geom/SSV/>).

Sistema d'E/S

Aquest sistema es va crear per establir un sistema fàcil i còmode per implementar i treballar amb dispositius d'entrada i de sortida. Un joystick, un sensor de força, un sistema hàptic (figura 4.4), sensors de posició, etc són dispositius comunament emprats en els laboratoris de robòtica i que fan que, en integrar-los dintre d'un software específic aquest sigui difícilment portable.



El sistema d'E/S proposa un conjunt de classes i interfícies a implementar per donar suport al simulador per nous dispositius sense que això impliqui que el codi es torni més rígid. Des del punt de vista del programa es pretén que el cicle i la forma d'adquirir les dades d'entrada siguin iguals independentment del dispositiu físic connectat.

Sistema sèrie

A vegades sol interessar que el simulador esdevingui el software de control del robot real. Això implica una comunicació entre el software i la controladora del robot. Aquest sistema proporciona suport per comunicar amb dispositius externs mitjançant un port sèrie (ja sigui per RS-232, pont USB, etc).

Com a funcionalitats interessants cal mencionar la possibilitat de fer comunicacions síncrones o asíncrones, definició de *timeouts*, configuració dels paràmetres de la comunicació (velocitat de modulació, bits de paritat, control de flux, etc...).

Aquest sistema té com a nucli central un CSerial, un projecte *Open Source* que proporciona una API còmoda per treballar amb els ports sèrie (<http://www.codeproject.com/KB/system/serial.aspx>).

Conclusions

Com s'ha vist l'ús de simuladors per validar dissenys és una opció molt útil, no només per validar dissenys, trobar errors i corregir-los sinó també a l'hora de fer programes per al robot. També permet paral·lelitzar

la fase d'implementació ja que es pot anar desenvolupant el programari del robot i provar-lo amb el robot virtual, l'altre que es va desenvolupant el prototip.

L'ús de simuladors també permet convertir un recurs escàs i no compartible, com és el robot físic, en un recurs disponible per tothom. Diverses persones poden treballar, fer proves i fer software amb el robot simulat gairebé tant bé com treballarien amb el robot real.

Capítol 5:

Seguiment intel·ligent de les eines

Introducció

Un element comú i important en qualsevol intervenció per laparoscòpia és l'ús d'una càmera encarregada d'enfocar la zona on treballarà el cirurgià. Actualment ha de ser una persona qualificada l'encarregada d'agafar la càmera i portar-la on el metge requereixi i aguantar-la en aquest lloc mentre es fa la intervenció. Hom pot apreciar el primer gran inconvenient d'una intervenció per laparoscòpia: Una persona qualificada desaproveitada agafant un estri, que ha de conèixer la intervenció que es realitzarà per situar-se de forma que no molesti al cirurgià i pugui dirigir la càmera on ell requereixi. Aquesta persona a vegades no ajuda tant com interfereix a la tasca del cirurgià. Sense voler el seu pols sol transmetre vibracions a la càmera que es tradueixen en inestabilitat a la imatge que està mirant el cirurgià.

Els sistemes de teleoperació i telepresència actuals han donat diverses solucions de tal forma que sigui el cirurgià que realitza la intervenció qui posicioni la càmera de forma còmode i, molts cops sense haver d'emprar les mans, destinades als estris quirúrgics.

Un cas concret és el del sistema *ZEUS* on el cirurgià controla la càmera amb comandes de veu. Una altre solució és la proposada per l'empresa *Prosurgics* amb el robot *FreeHand* (figura 5.1), dissenyat per portar la càmera i comandat de forma gestual amb el cap.

La major part de solucions actuals continuen requerint una actuació directa per part del metge. S'ha eliminat un assistent afegint una càrrega extra al cirurgià, que ara ha de controlar la càmera a més de les eines. En conclusió, les solucions actuals carreguen de feina i preocupacions al cirurgià en comptes d'ajudar-lo per que només s'hagi de preocupar de la intervenció.

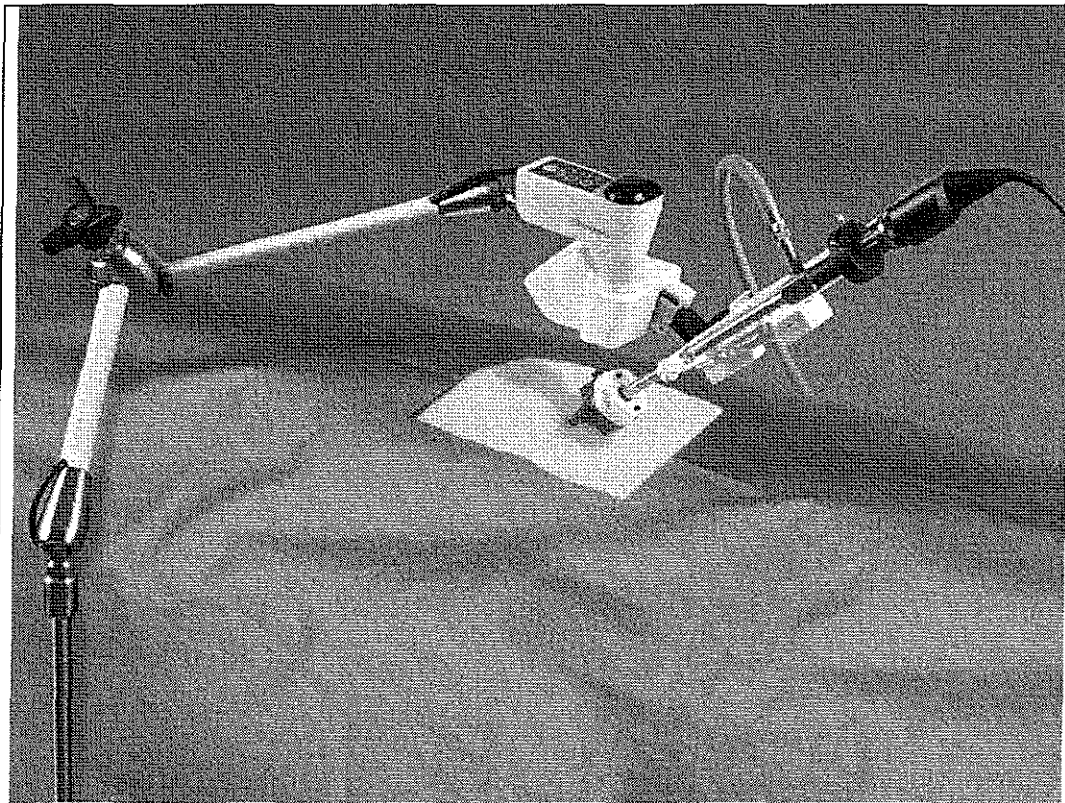


Figura 5.1: El robot FreeHand pensat per portar una càmera de forma gestual

Per donar una solució real al problema del control de la càmera el sistema pòrtic de 3 braços incorpora un sistema intel·ligent per portar-la allà on el metge requereixi. A més el metge pot ajustar certs paràmetres de seguretat com és la distància de la càmera a la zona de treball.

Donada la importància d'aquest estudi es va aprofitar l'existència d'un robot pòrtic d'un braç (figura 5.2) per acabar de provar les estratègies de seguiment. Un cop les proves amb el simulador

eren prou satisfactòries es connectava el robot per a veure si s'observava el mateix comportament que amb el simulador.

Quan les proves van ser concloents el codi final es va adaptar i es va incorporar al simulador de 3 braços.



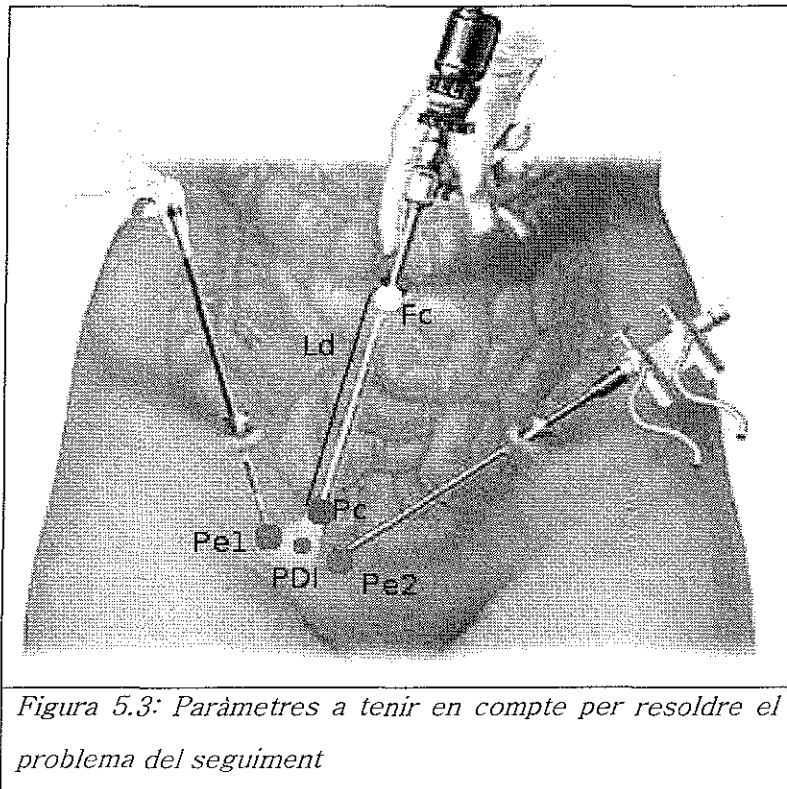
Figura 5.2: Robot pòrtic d'un braç emprat per provar el sistema de seguiment

Estratègia de seguiment

Per donar una solució correcta al problema del seguiment s'ha d'entendre bé el context d'una intervenció a fi de formalitzar el problema. En una intervenció per laparoscòpia les eines i la càmera pivoten sobre els seus punts de fulcre. De fet l'únic punt de fulcre que s'ha de considerar és el de la càmera F_c ja que imposa una restricció a la solució final. Altres factors a tenir en compte són els punts a l'espai dels

extrems de les eines $Pe = \{Pe_1, \dots, Pe_n\}$ i una distància D que determina el zoom de la imatge. Finalment s'ha de conèixer el punt on s'ha de posicionar l'extrem de la càmera Pc (figura 5.2).

Tenint en compte que l'objectiu final es moure el braç robot cap a un punt Pb tal que aquest posicioni la càmera al lloc desitjat s'ha de conèixer també la longitud del laparoscop Ld .



La primera part del problema és definir el punt que ha d'enfocar la càmera en funció dels punts $Pe = \{Pe_1, \dots, Pe_n\}$. Dit d'una altre forma: s'ha de definir una funció $Focus(P_1, \dots, P_n)$ de tal forma que

$$PDI = Focus(Pe_1, \dots, Pe_n)$$

De fet, per variar l'estratègia únicament s'ha de canviar la funció per que retorni un altre punt. Com a cas principal d'estudi s'ha fet servir la funció de punt mig.

$$PDI = Focus(Pe_1, \dots, Pe_n) = \frac{\sum_{i=1}^n Pe_i}{n}$$

Els casos més trivials serien els seguiments dels extrems de cada eina de forma independent.

$$PDI = Pe_i$$

La segona part del problema és trobar el punt de l'extrem de la càmera Pc . Aquest punt es troba sobre la recta que passa per PDI i el punt de fulcre Fc i té com a vector director

$$\vec{v} = \frac{\overrightarrow{PDIFc}}{|\overrightarrow{PDIFc}|}$$

A més s'ha de complir que la distància entre PDI i Pc ha de ser D sempre i quan això no impliqui que Pc estigui fora del cos. Amb aquestes consideracions es conclou que

$$\text{Si } dist(PDI, Fc) > D$$

$$Pc = PDI + D\vec{v}$$

$$\text{Sinó } Pc = Fc$$

$$\text{Fsi}$$

Per finalitzar, s'ha de trobar el punt on s'ha de posicionar el braç per que l'extrem de la càmera es situï a P_c . Si la longitud de l'eina és L_d el braç s'ha de col·locar a P_b .

$$P_b = P_c + L_d \vec{v}$$

A partir d'aquest punt el posicionament del braç a P_b és responsabilitat del càlcul cinemàtic del robot. Aquesta delegació de responsabilitat permet que les estratègies de seguiment siguin independents de l'arquitectura del robot de forma que el mètode de seguiment pot ser implementat amb qualsevol robot cartesià del mercat.

Per acabar s'ha de tenir en consideració que en una intervenció real el cirurgià està movent les eines contínuament, la càmera no. D'alguna forma el sistema s'ha de moure realment quan el cirurgià vulgui canviar la seva zona de treball. És per això que el sistema permet definir un radi d'histeresi H tal que la càmera només es mou si el PDI surt de la zona definida per aquest radi.

Si $H > \text{dist}(PDI_i, PDI_{ref})$ llavors

$\text{moureCamera}(PDI_i, F_c, D, L_d)$

$$PDI_{ref} = PDI_i$$

Fsi

El simulador d'un braç

Per portar a terme l'estudi del sistema de seguiment es va optar per la dinàmica de simular abans de provar amb el robot real. Aquesta decisió es va prendre en part per la necessitat de poder treballar en

paral·lel amb l'equip que estava treballant en el disseny i programació de la controladora del robot.

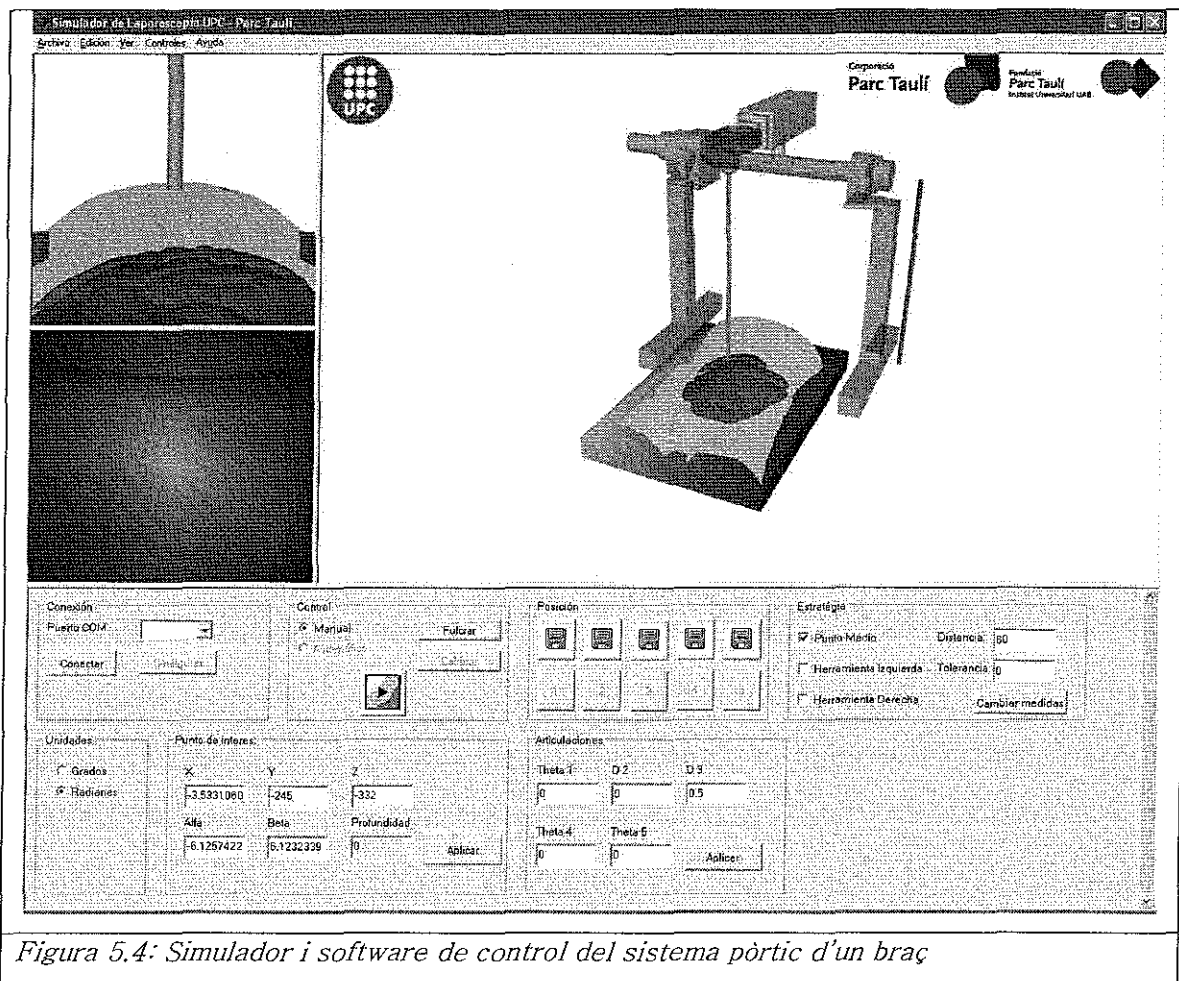


Figura 5.4: Simulador i software de control del sistema pòrtic d'un braç

El simulador d'un braç (figura 5.3) és una versió adaptada del simulador de 3 braços. A causa de les grans semblances entre sistemes, l'adaptació només va requerir el canvi dels models 3D i les equacions de cinemàtica. Es va incloure codi per llegir sensors de posicionament i una interfície per fer comunicacions via RS232 per comunicar amb la placa controladora. Finalment aquest simulador va esdevenir el software de control actual del robot pòrtic d'un braç.

Implementació i proves

Un cop el simulador y la controladora del robot estaven prou avançades van començar les proves del sistema en conjunt. El simulador s'encarregava del càlcul cinemàtic i enviava les configuracions de les articulacions a la controladora. Aquesta s'encarregava de fer el correcte desplaçament dels motors per assolir la configuració demanada. Un detall que li queda pendent a la controladora és la lectura dels encoders dels motors a fi de poder donar una retroalimentació de la posició del robot. De moment es treballa amb la hipòtesi de que els motors són sempre a la posició que li dóna el simulador.

Les eines van ser sensoritzades amb un sistema de posicionament magnètic (*Polhemus*) a fi de poder saber la posició del seu extrem. Amb un procés de cal·libració inicial s'establia la relació entre el sistema de coordenades del robot i el sistema de coordenades del *Polhemus* a fi de determinar la posició de les eines en el sistema de coordenades del robot.

Aquesta cal·libració inicial estableix també les posicions inicials de les eines. A partir d'aquest punt les següents posicions es calculen de forma incremental per minimitzar l'error associat a l'entorn (materials elèctrics, metalls, etc) La demostració matemàtica d'aquesta minimització es la següent:

Sigui

$$C_i = L_i + e_i$$

la captura i -èsima del *Polhemus* la qual es pot veure com la lectura de la posició L_i amb un error e_i donat per l'entorn. Si considerem que les

variacions de l'entorn són mínimes durant una intervenció podem preveure que

$$e_1 \approx \dots \approx e_n \approx e$$

Del que es conclou que:

$$\Delta e_i = e_i - e_{i-1} \approx e - e = 0$$

Si les posicions de les eines es calculen de forma incremental tenim que:

$$P_i = P_{i-1} + \Delta C_i$$

$$P_i = P_{i-1} + \Delta L_i + \Delta e_i$$

$$P_i \approx P_{i-1} + (L_i - L_{i-1}) + 0$$

$$P_i \approx P_{i-1} + (L_i - L_{i-1})$$

El resultat: és pot determinar la posició a l'espai de les eines minimitzant l'error del *Polhemus* associat a l'entorn. Finalment s'ha de posar molta cura en el cal·librat i establiment de la posició inicial ja que aquest error s'anirà arrossegant durant tot el moviment.

Les proves principals que es van portar a terme consistien en introduir les eines i la càmera portada pel robot dintre d'una maqueta a escala real d'un tors humà i anant simulant l'accés als diferents òrgans. La finalitat principal era reproduir les intervencions per laparoscòpia més comuns i veure com es comportava el sistema.

Per acabar només comentar que aquest sistema robòtic va ser el projecte elegit per la UPC per ser exposat a la fira de recerca *Recerca en directe 08* organitzat pel *Parc Científic de Barcelona* (figura 5.5).

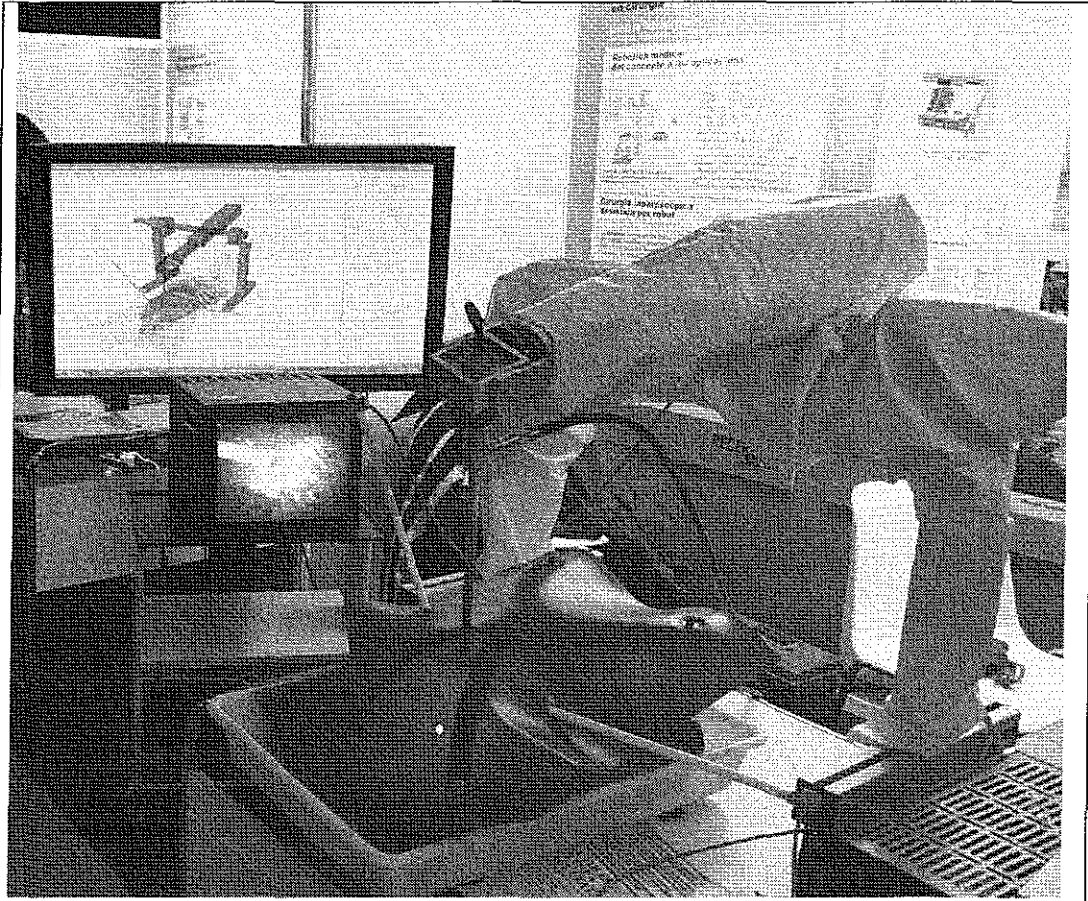


Figura 5.5: Estant de la fira Recerca en directe 08 mostrant el sistema robòtic de seguiment d'eines

Conclusions

En aquest capítol s'ha explicat el procediment complet d'elaboració i assaig de les estratègies de seguiment. Partint d'una formalització matemàtica del problema i dels elements implicats va ser fàcil trobar les relacions entre ells i elaborar una solució.

La següent part era trencar les diferències entre els plantejaments teòrics i el món real on els aparells de mesura tenen un error. Formalitzar matemàticament aquest fet va permetre elaborar un sistema de tractament de les dades dels sensors per minimitzar l'error.

Un cop assajats i provats aquest mètodes amb un robot real el codi final es va afegir al simulador de 3 braços. De cara a un robot real s'ha de tenir en compte que la posició de les eines ja no s'obtindran amb sensors sinó que el robot la sap en tot moment amb un error més petit que el que s'obtenia amb el sistema de posicionament.

Capítol 6:

Resultats finals

Introducció

Un cop desenvolupat el simulador es passa a validar el sistema amb aquesta eina. Aquesta validació té dos aspectes: una avaluació mèdica i una avaluació tècnica. L'avaluació mèdica es du a terme simulant diversos escenaris proposats pels cirurgians on aquest provaran el robot simulat per comprovar si els hi dóna tota la llibertat i abast necessari.

L'avaluació tècnica defineix unes dimensions inicials del robot i estudia el comportament matemàtic del sistema. Un aspecte important d'aquest estudi serà comprovar la precisió de les equacions, acotar-ne l'error i estudiar el seu comportament. S'ha de tenir en compte que si el càlcul de la cinemàtica del robot no és prou precís s'haurà de trobar altres mètodes per obtenir-la.

L'avaluació mèdica s'encarrega de refinar les dimensions inicials i de classificar els àmbits d'ús del sistema. Aquesta classificació conclourà en quines intervencions hauria d'intervenir la nova plataforma i per a quines no és idònia. Aquesta avaluació proporcionarà molta informació útil per a l'estudi de noves plataformes que puguin ser emprades en qualsevol intervenció per laparoscòpia.

En aquest capítol es tractaran principalment els resultats de l'avaluació tècnica i es comentaran certes proves de l'avaluació mèdica.

Avaluació tècnica

Aquesta avaluació té dos objectius principals:

1. Trobar unes mides del robot inicialment vàlides i que després s'acabaran d'ajustar amb els resultats de l'avaluació mèdica.
2. Estudiar la precisió de la cinemàtica per a veure si compleix els requeriments.

Estudi de les mides inicials del robot

Per a definir aquestes mides es va modelar un tors humà d'una persona adulta de 1.80 metres d'alçada. Aquest model es va incloure al simulador amb la mateixa escala que el robot. Considerant una longitud promig de 30 cm de llargada per les eines de laparoscòpia i 40 cm per la càmera es varen fer proves per fixar unes dimensions inicials prou encertades.

Es resultats es resumeixen a la següent taula:

Alçada mínima del pòrtic	40 cm
Alçada màxima del pòrtic	70 cm
Recorregut de la guia del pòrtic	50 cm
Abatiment del braç	-50º, 25º
Longitud mínima del braç	40 cm
Longitud màxima del braç	70 cm

Estudi de la precisió de la cinemàtica

Aquest estudi té com a objectiu comprovar la precisió de l'estudi cinemàtic. Per la part de la cinemàtica directa s'ha de tenir en compte que el formalisme de Denavit-Hartenberg permet resoldre la cinemàtica molt fàcilment però que pot acumular un error més gran que una obtenció de la cinemàtica directa per mètodes geomètrics purs. La raó ve determinada pel fet que cada grau de llibertat implica un producte matricial on cada matriu té 4 elements que són funcions trigonomètriques (sin i cos). Computacionalment aquestes funcions tenen precisió finita i el fet de que participin de productes matricials fan que l'error vagi augmentant a mesura que es propaga.

La cinemàtica inversa tractada amb mètodes geomètrics també pot presentar greus imprecisions si hi ha una forta composició de funcions on hi participin moltes funcions trigonomètriques. Com a norma general les solucions més petites i simples solen ser més precises.

De fet la resolució de la cinemàtica inversa que es va implementar va ser la segona solució que es va trobar ja que la primera versió (mes llarga i complexa) acumulava un error intolerable quan $\theta_6 \rightarrow 0$.

Per desgracia no es pot estudiar l'error per separat de cada part (IK i DK) ja que no es té una taula que relacioni configuracions de les articulacions amb posicions finals obtingudes amb un mètode 100% precís. De fet tampoc és greu no poder-los estudiar per separat ja que l'algorisme principal del moviment del robot els feia servir de manera conjunta:

1. Obtenir posició final desitjada P

2. Obtenir configuració del robot per assolir-la:

$$\langle \sigma_1, \dots, \sigma_n = IK(P) \rangle$$

3. Aplicar aquesta configuració per arribar a P

$$P = DK(\sigma_1, \dots, \sigma_n)$$

Com que IK i DK són funcions oposades es pot trobar l'error del procediment anterior fent servir la propietat que té la composició de funcions en el cas d'una funció i la seva inversa:

$$P = (f \circ f^{-1})(P)$$

Però IK i DK no són funcions precises per la qual cosa la relació es modifica de la següent forma:

$$\begin{aligned} P &\approx (DK \circ IK)(P) \Rightarrow \\ P &= (DK \circ IK)(P) + \vec{e} \end{aligned}$$

On \vec{e} serà el vector que representa l'error acumulat i s'obté fàcilment de l'equació anterior com a:

$$\vec{e} = P - (DK \circ IK)(P)$$

De cara a estudiar el comportament de la precisió és més convenient estudiar l'error com la distancia entre el punt desitjat i el punt on s'ha arribat.

$$e = \text{dist}(P, (DK \circ IK)(P)) = |\vec{e}|$$

Un cop definit el quantificador de l'error es procedeix al seu estudi. Per això es farà un anàlisi estadístic sobre una població de punts de la zona de treball del robot generats aleatòriament.

El procediment a seguir és el següent:

1. Generar una mostra de N punts de la zona de treball del robot.
2. Per cada punt P_i de la mostra trobar la configuració del robot amb la cinemàtica inversa.
3. Calcular la posició P'_i del robot amb la configuració trobada en el pas 2.
4. Calcular l'error e_i com a $e_i = \text{dist}(P_i, P'_i)$

5. Obtenir la mitjana com a $e_m = \frac{\sum_{i=1}^N e_i}{N}$ i la desviació estàndard

$$\text{com a } e_d = \sqrt{\frac{\sum_{i=1}^N (e_i - e_m)^2}{N-1}}$$

A continuació es mostren els estadístics i els histogrames de diferents poblacions amb $N=100$ (figura 6.1), $N=1000$ (figura 6.2) i $N=10000$ (figura 6.3). Per problemes de precisió simple les mostres s'han escalat per un factor 10^8 , ja que el software estadístic del que es disposa no pot treballar amb valors tant petits com els resultats obtinguts.

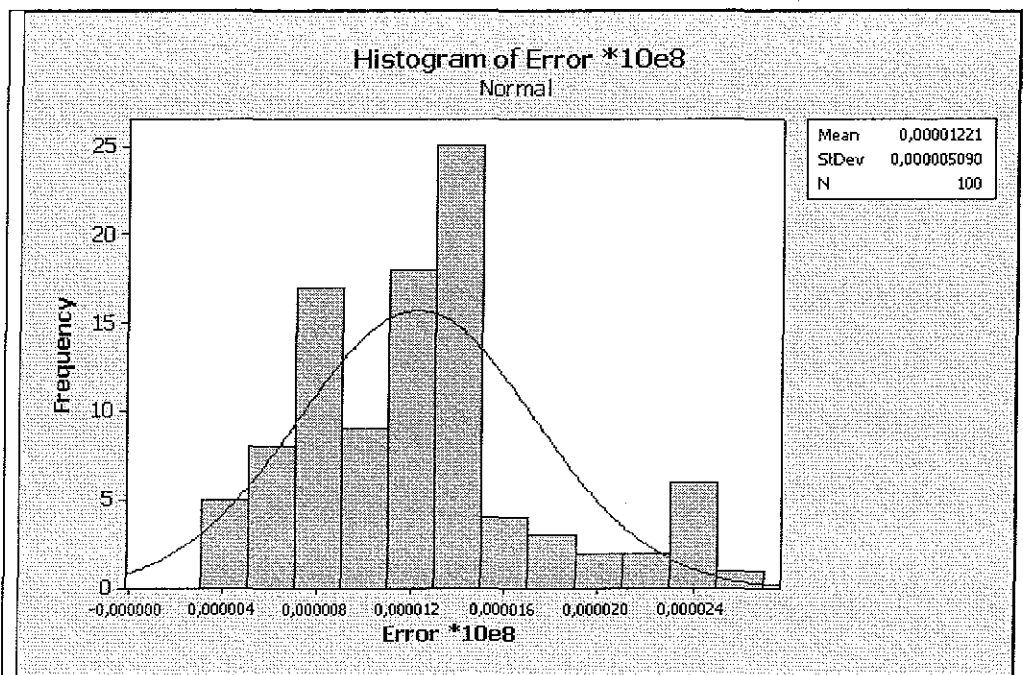


Figura 6.1: Resultats obtinguts per una població N=100

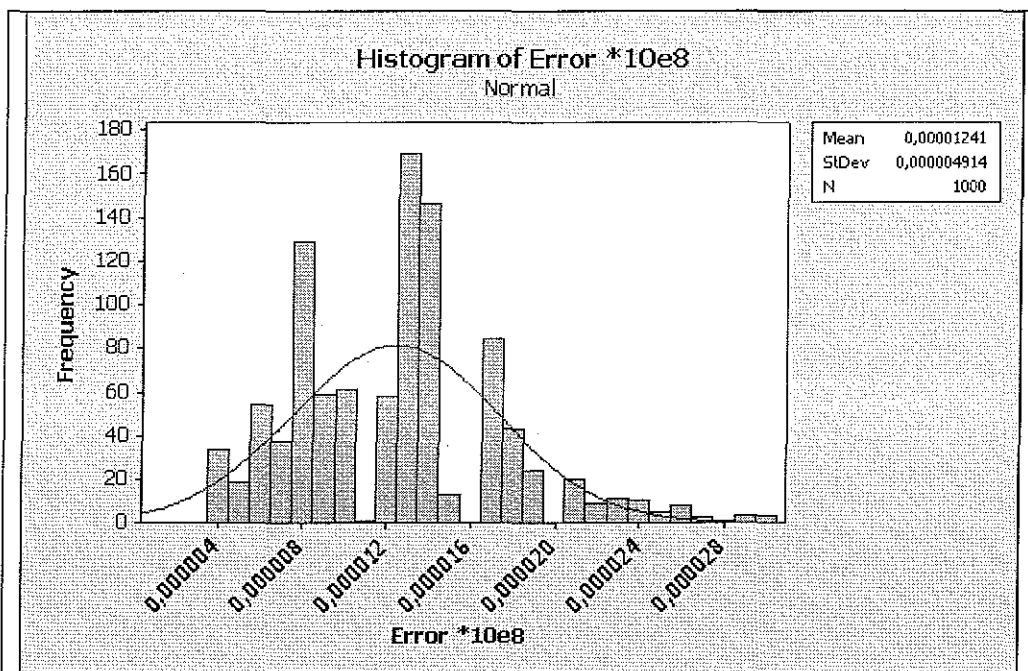
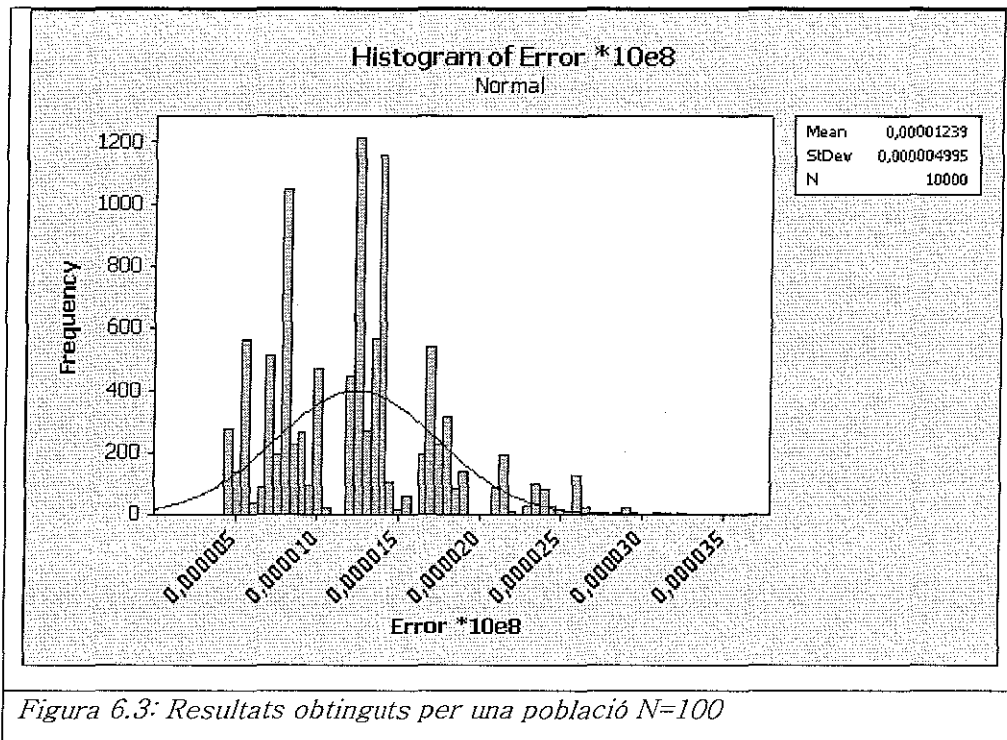


Figura 6.2: Resultats obtinguts per una població N=1000



Si ara recuperem l'escala original els resultats es resumeixen a la següent taula:

Punts de la mostra	Mitjana	Desviació estàndard
100	1.22e-13 mm	5.09e-14 mm
1000	1.24e-13 mm	4.91e-14 mm
10000	1.23e-13 mm	4.95e-14 mm

El primer resultat a tenir en compte és que per totes tres poblacions la precisió es troba per sota del picometre (10^{-12}), una precisió més que suficient per les tasques que es volen desenvolupar. A la vista d'aquest resultats es conclou que no cal destinar més esforços a l'estudi de l'error i millora de la precisió de l'algorisme.

Avaluació mèdica

Per a portar a terme aquesta avaluació es van proposar un conjunt d'intervencions per a ser simulades. L'èxit de cada simulació es pondera per factors addicionals com poden ser la demanda de la intervenció, la dificultat i el perill que comporti pel pacient.

Per exemple, una simulació amb èxit d'una apendicitis puntarà positivament en quant a demanda (actualment totes es fan per laparoscòpia). No puntuarà gaire per dificultat i puntuaria moderadament per a risc del pacient (el teixit infectat s'ha d'extreure amb cura, sense que caigui a l'interior del pacient). D'aquesta forma es va donar una puntuació qualitativa del sistema des de un punt de vista mèdic.

Aquestes simulacions van permetre refinar les mides finals del braç robot, que queden de la següent manera:

Alçada mínima del pòrtic	40 cm
Alçada màxima del pòrtic	70 cm
Recorregut de la guia del pòrtic	60 cm
Abatiment del braç	-50º, 25º
Longitud mínima del braç	40 cm
Longitud màxima del braç	80 cm

Finalment les simulacions d'intervencions van concloure que el robot podia efectuar un gran nombre d'intervencions quan les eines i la càmera accedien per la part central de l'abdomen. El punt feble del robot el varen trobar per intervencions on s'accedia a l'interior del cos per la

part lateral (costelles, fetge, etc...) o amb accessos de les eines massa inclinats (accés per la zona pèlvica amb inclinacions properes als 90°).

Noves propostes d'investigació

Les plataformes actuals parteixen d'un paradigma d'un robot amb n braços. El problema principal és que a mesura que guanyen popularitat i són emprades per més metges aquest demanen més braços. Un exemple actual és el robot *Da Vinci* que inicialment tenia 3 braços i actualment ja en té 4 (figura 6.4) sense possibilitats d'afegir-ne un 5è.

El robot pòrtic de 3 braços està pensat única i exclusivament per 2 eines i la càmera. La disposició i els moviments dels braços fan la seva extensió inviable. En resum, la seva arquitectura és molt tancada. Les futures investigacions han de complir aquest requeriment d'adaptabilitat

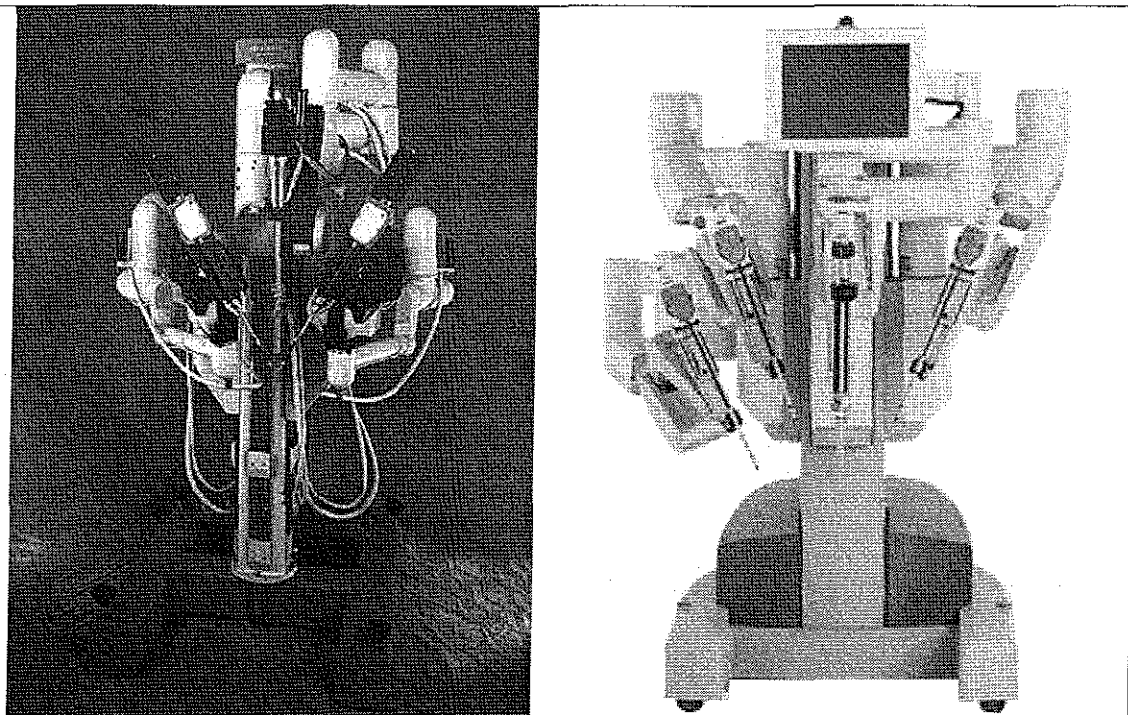


Figura 6.4: evolució del sistema Da Vinci de 3 braços a 4.

El problema trobat amb els robots amb múltiples braços ha donat lloc a l'estudi de noves plataformes quirúrgiques basades en robots independents d'un braç treballant de forma col·laborativa. Aquest robots es poden col·locar a diferents llocs del camp operatori de forma que el conjunt d'operacions on es poden fer servir s'amplia de forma considerable (figura 6.5). Un problema derivat d'aquest nou paradigma és la necessitat d'un sistema de localització dels braços a fi d'obtenir les seves coordenades i fer els càlculs necessaris per les intervencions.

Aquest nou paradigma permet variar el nombre de braços que hi intervindran com la posició inicial dels braços, dos variables a tenir en compte a l'hora de planificar una operació i que podien ser determinants per concloure si una intervenció es podia realitzar amb un sistema robòtic o no. Amb aquest guany de llibertat es pot adaptar el sistema al gran ventall d'intervencions que es realitzen per laparoscòpia.

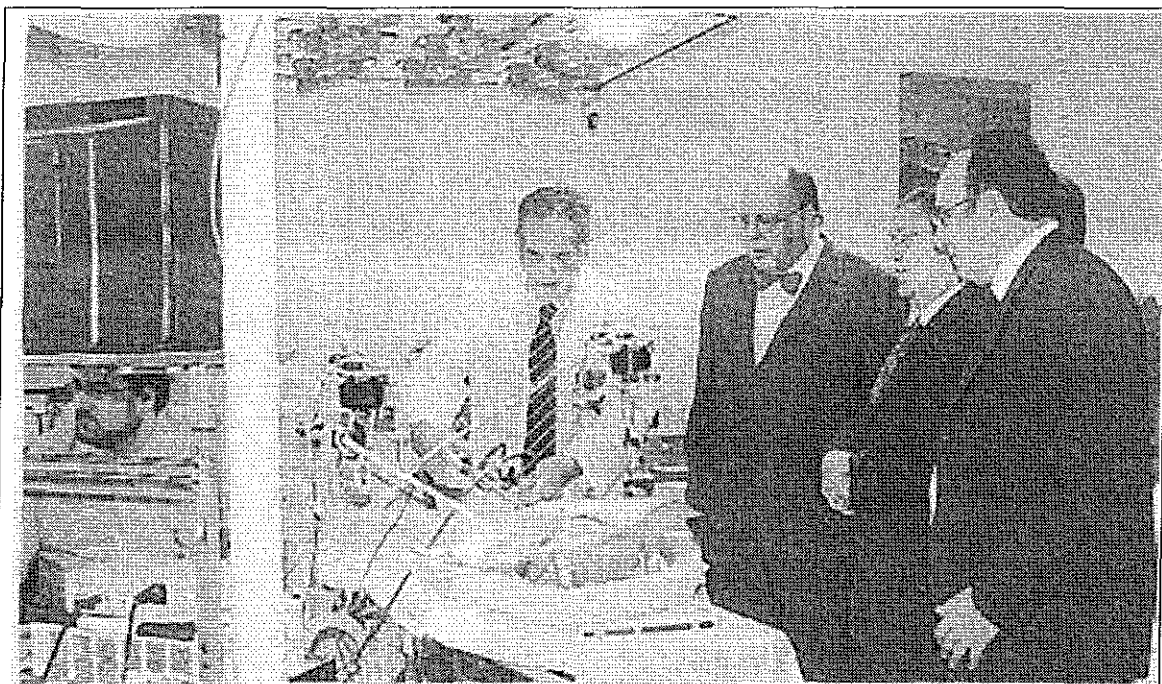


Figura 6.5: Fotografia del Diari de Sabadell del 20 de Desembre del 2008 mostrant la investigació amb robots cooperatius.

Conclusions:

Després de gairebé un any de treball dedicat al desenvolupament d'una nova plataforma robòtica assistencial i, fent un recull de la feina realitzada, s'observa que el procés de desenvolupament ha generat nous procediments i eines per a l'estudi de noves plataformes. Fent una recapitulació:

1. S'ha provat un nou model d'investigació basat en la simulació. Aquest model permet treballar i experimentar amb noves arquitectures robòtiques sense tenir un prototipus real. D'aquesta forma el cost de la fabricació d'un prototip es retarda fins a les darreres fases d'investigació, garantint la inversió.
2. S'ha estudiat i implementat un sistema de seguiment intel·ligent d'eines independent del robot. De forma que es pot adaptar a d'altres plataformes robòtiques. Aquest estudi és pioner al món ja que no hi ha altres sistemes comercials que ho facin.
3. S'ha obtingut un software fàcilment adaptable a noves plataformes robòtiques d'estudi. De cara a noves investigacions el temps i el cost de desenvolupament d'un simulador es redueixen dràsticament.
4. S'han obert nous projectes d'investigació amb robots cooperatius aprofitant els èxits d'aquest projecte i prenent de les mancances

La bona investigació és aquella que va recollint el testimoni del que altres varen fer i, de la mateixa manera, la deixen a disposició d'altres

investigadors. L'herència d'un investigador no són només els èxits sinó també els fracassos. Els èxits permeten avançar, els fracassos diuen per on.

Si tenim en compte tot això es conclou que el projecte ja ha donat fruits molt abans de tenir un producte final.

Capítol 7:

Pressupost del sistema

Introducció

Per a concloure el desenvolupament del sistema s'han d'elaborar un pressupost de producció. Aquest s'ha d'obtenir a partir del cost del desenvolupament del prototip i un estudi de rendibilitat per obtenir el nombre d'unitats convenient per cobrir els costos de desenvolupament.

La correcta elaboració del pressupost de producció és una tasca molt crítica. Un pressupost malament elaborat pot generar pèrdues molt greus a les empreses que poden portar-les, fins i tot, a la ruïna.

Desenvolupament del prototip

A continuació es detalla el cost de desenvolupament del prototip. Aquest s'elabora principalment a partir del planing del projecte (vist al capítol 2) on es reflecteixen les durades de cada tasca i el personal assignat.

Els costos del material per la fabricació del prototipus surten de decisions tècniques, com són les dimensions del robot i decisions mèdiques reglamentades com són el tipus de material que s'ha d'emprar en la fabricació d'equipament quirúrgic. Els materials per a la construcció del robot són el ferro i el titani, principalment per la seva duresa i el fet de que són hipoal·lèrgics.

La següent taula reflexa el cost total de desenvolupament del prototip:

Motiu	Personal	Dies	Hores per dia	Nombre de persones	Sou x hora	Cost
Disseny mecànic:	Enginyer industrial	40	8	1	45	14.400
Disseny simulador 3D:	Enginyer informàtic	210	8	1	45	75.600
Simulació i avaluació:	Enginyer informàtic	10	8	1	45	3.600
	Cirurgia	5	4	3	60	3.600
Material del prototip:						18.400
Muntatge i proves:	Enginyer industrial	30	8	2	45	21.600
Desenvolupament de la unitat de control:						17.500
Desenvolupament del software de control:	Enginyer informàtic	6	8	1	45	2.160
Desenvolupament del software d'autoverificació i ajut al muntatge:	Enginyer informàtic	35	8	1	45	12.600
Cost total del desenvolupament del prototip:						169.460

D'aquest pressupost cal remarcar dos punts:

1. El cost del desenvolupament del software de control és reduït perquè aquest software aprofita molt de codi del simulador.
2. Es desenvolupa un software d'autoverificació i ajut al muntatge per a reduir els costos de fabricació, instal·lació i muntatge per unitat.

Estudi de rendibilitat

A l'hora de decidir quants sistemes i a quin preu s'haurien de vendre per amortitzar el cost del desenvolupament s'ha de tenir en compte diversos factors, que responen a un anàlisi DAFO tant del

producte com de les empreses implicades. A continuació es destaquen els punts mes importants:

1. Debilitat:

Aquest sistema es va pensar per a ser emprat en hospitals i clíniques especialitzades en cirurgia laparoscòpica o que el seu volum d'intervencions justifiqués la compra d'un equip assistencial tant complex.

2. Amenaces:

Tot i que les intervencions assistides per robot s'estan popularitzant cada cop més, la realitat es que hi ha molt poques empreses que s'hi dediquin plenament amb productes tant complex. De fet el mercat actual es troba una mica monopolitzat per *Intuitive Surgical* i la seva família de productes. Molts d'aquests projectes varen ser desenvolupats per tercers a qui varen comprar les patents i els drets de producció i explotació.

3. Fortalesa:

El sistema ha sigut desenvolupat conjuntament per la U.P.C. i el Parc Taulí, dos institucions de reconeguda fama mundial pels seus estudis i avenços. Aquest dos segells avalen el sistema i li confereixen fiabilitat donant més confiança a futurs compradors i inversors.

4. Oportunitat:

Les intervencions amb robots s'estan popularitzant cada cop més i les poques empreses que s'hi dediquen actualment no s'arrisquen a treure un producte que pugui crear competència amb el *DaVinci*. El sistema pòrtic té unes prestacions que poden fer-li front.

Considerant que el mercat potencial és reduït s'ha de pensar que els veritables beneficis del sistema es troben amb la comercialització del suport i serveis complementaris, principalment els serveis de manteniment. Si la garantia inicial es d'un any els serveis addicionals de manteniment es començaran a contractar a l'any de la venda d'una unitat.

La conclusió principal és que els costos d'investigació s'han de cobrir únicament amb la venda i instal·lació de les primeres unitats.

Pressupost de producció

A continuació es detalla el preu de venda de les primeres unitats, que es preveuen que es puguin vendre el primer any de comercialització del producte. Es consideren els cassos de 8 i 12 unitats el primer any:

Motiu	Cost x 8	Cost x 12
Matèries/unitat:	13.250,0	13.100,0
Muntatge i verificació:	8.400,0	8.400,0
instal·lació i posada en marxa:	1.200,0	1.200,0
Repercussió del desenvolupament:	21.182,5	14.121,7
Documentació, assistència tècnica i garantia:	3.300,0	3.300,0
Cost per unitat:	47.332,5	40.121,7
Cost indirectes empresa:	1.550,0	1.550,0
Cost net per unitat:	48.882,5	41.671,7
Beneficis (15%):	7.332,4	6.250,8
Preu de venda:	56.214,9	47.922,4
I.V.A (16%)	8.994,4	7.667,6
Preu final de venda:	65.209,3	55.590,0

Conclusions

L'objectiu principal d'aquest capítol és trobar el preu final de venda del sistema a fi de que les primeres unitats cobreixin el cost de desenvolupament i proporcionin un petit benefici. Un cop s'ha cobert el cost de desenvolupament el marge de beneficis augmenta substancialment i permet la financiació de noves investigacions.

El preu de venda final té un gran atractiu pel mercat, sobretot si es té en compte que el sistema *DaVinci* té un cost de venda que oscil·la entre el milió i el milió i mig de dòlars.

Bibliografia

Llibres de consulta

- John J. Craig, Robótica, tercera edición, Pearson Education, 2006.
- Antonio Barrientos, Luís Felipe Peñín, Carlos Balaguer, Rafael Aracil, Fundamentos de robótica, segunda edición, Mc Graw Hill, 2007
- Joan Trias Pairó, Geometria per la informàtica gràfica i CAD, Edicions UPC, 1999
- Bjarne Stroustrup, El lenguaje de programación C++, Addison Wesley, 2002
- Ron Fosner, OpenGL Programming for Windows 95 and Windows NT, Addison Wesley, 1997
- Mason Woo, Dave Shreiner, Jackie Neider, Tom Davis, OpenGL® Programming Guide: The Official Guide to Learning OpenGL®, 5th edition, Addison Wesley, 2005
- Mercè Galán, Blender: Curso de iniciación, Inforbook's Ediciones, 2007

Webs de consulta

- Intuitive Surgical Inc: <http://www.intuitivesurgical.com/>
- Roboop Home Page: <http://www.cours.polymtl.ca/roboop/>
- Centro de desarrollo de Visual C++:
<http://msdn.microsoft.com/es-es/visualc/default.aspx>
- Blender.org Home: <http://www.blender.org/>

- Python Programming Language – Official Website:

<http://www.python.org/>

- OpenGL The Industry's Foundation for High Performance Graphics:

<http://www.opengl.org/>

Annex I: Eines per la importació/exportació de models geomètrics des del software de disseny al software de simulació

Introducció:

Un element comú que té tot software 3D és el fet que ha de mostrar models pre dissenyats amb un software de modelat 3D. Molt sovint aquest software genera més informació de la desitjada per la visualització. A més molts formats de fitxers 3D són propietaris i/o incompatibles amb l'aplicació que s'està desenvolupant. És per això que els desenvolupadors prefereixen especificar el seu format propi i intern i fer eines per traduir d'un format a un altre. De fet, la tendència actual és que els software de modelat incorporin algun sistema de programació mitjançant *scripts* per poder aplicar algorismes directament sobre els models. Amb aquests mecanismes de programació d'*scrips* es pot fer software d'importació/exportació més fàcilment.

El software de modelat escollit per aquest projecte va ser *Blender* (figura A1.1) (www.blender.org) ja que és lleuger (10 MB) multiplataforma i *Open Source*. A més fa servir el llenguatge *Python* (www.python.org) que es un llenguatge d'*scripting* independent de *Blender*, lliure i molt potent, fins al punt d'incorporar orientació a objectes. La seva sintaxi és còmoda i simple i té un tractament molt intuïtiu per conjunts i iteradors.

Un altre motiu per recomanar *Blender* és el fet d'incorporar un bon conjunt d'importadors i exportadors als formats més comuns de fitxers 3D i la facilitar per trobar-ne d'altres per internet.

El format intern que es fa servir al simulador ve marcat per l'ús de la llibreria RPQ (llibreria interna del departament ESAII) on l'única informació que es guarda són els triangles d'un objecte. Unes ampliacions posteriors van permetre el fet de poder guardar textures per objecte i normals per vèrtex, necessaris per una millor visualització.

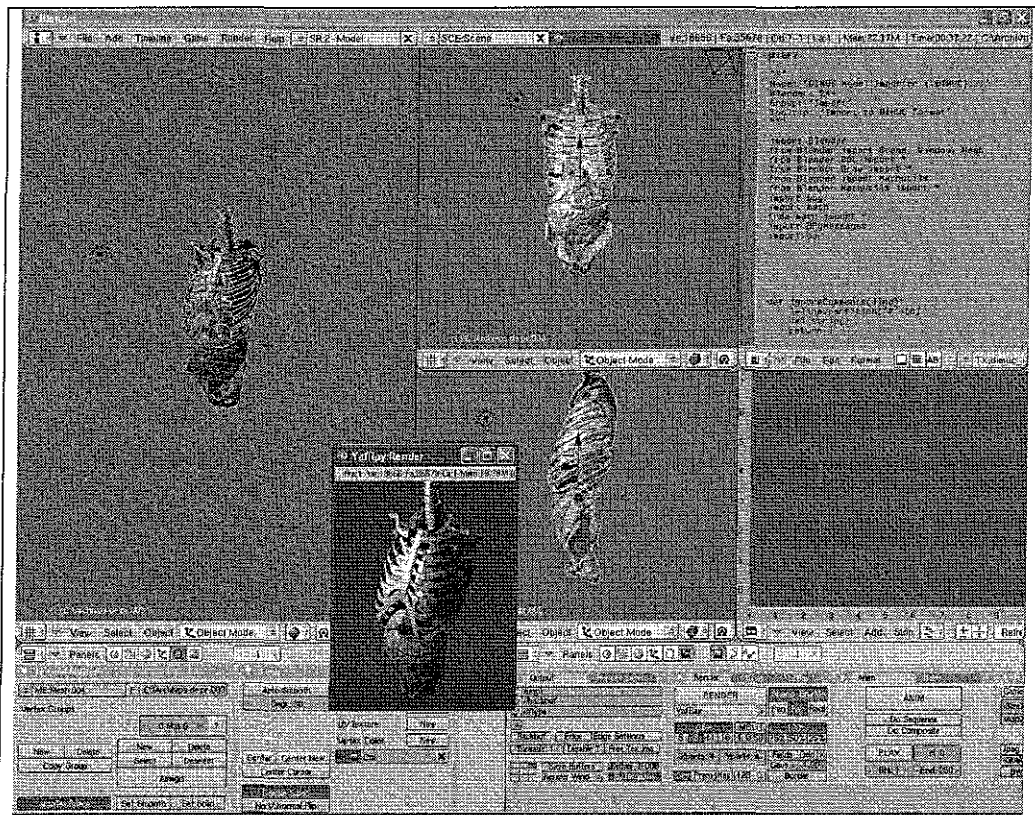


Figura A1.1: Blender en funcionament

Especificació del format RPQ (.tris)

Els fitxers d'informació gràfica de RPQ s'identifiquen amb una extensió .tris i contenen la informació d'una malla 3D (normalment

associada a un element mòbil del robot) (figura A1.2). El format és molt simple. La primera línia del fitxer conté el nombre de cares de la malla i, a continuació s'escriu la informació de cada triangle per pàrrafs. Cada pàrraf conté tres línies amb els valors x, y, z (en mil·límetres) del punt del triangle. Aquest punts s'han d'escriure en sentit antihorari per poder calcular un vector normal correcte per la il·luminació.

Aquesta simplicitat en el format té les seves coses a favor i les seves en contra. Com a principal avantatge és que la càrrega del model és molt ràpida ja que només ha de llegir de fitxer. Com a desavantatge podem veure que es guarda molta informació redundant (punts comuns a diversos triangles apareixen diversos cops) a més de pèrdua d'informació que podria ser molt útil.

7844		
-56.41	-50.41	431.1
-53.8	-46.73	424.8
-46.01	-56.94	418.4
-56.41	-50.41	431.1
-46.01	-56.94	418.4
-49.86	-60.05	425.8
-12.56	80.78	463.9
-32.15	76.12	462.8
-65.35	48.42	477.1
...		
Figura A1.2: Extracte d'un arxiu .tris		

Eines

Les eines d'importació/exportació inclouen 4 *scripts* per *Blender* i un programa: un importador i un exportador de models .tris, un

importador i un exportador d'escenes. DIMUC i un programa per generar els fitxers . tris a partir d'un fitxer d'escena .DIMUC.

Factor de conversió

Donat que els programes de modelat no treballen en unitats de longitud reals com poden ser metres o centímetres és necessari especificar un factor de conversió a l'hora d'importar i exportar models. En aquest cas si s'importa un model .tris a *Blender* amb un factor 1 es veu un escalat desorbitat. El factor recomanable és 100. També s'ha de tenir en compte fer servir el mateix factor per exportar.

Coordenades globals i locals

Els programes de modelat 3D defineixen un sistema de coordenades globals i una matriu de transformació per cada malla. Les coordenades dels vèrtex d'una malla s'expressen relatives a la seva matriu a fi de que certes transformacions com poden ser rotacions, translacions i escalats és facin manipulant la matriu en comptes d'haver de manipular tots els vèrtexs de la malla.

Importador/Exportador .tris

L'importador d'arxius .tris carrega una malla des d'un fitxer .tris interpretant les seves coordenades com coordenades globals és a dir, relatives al sistema de coordenades global. L'exportador d'arxius .tris

guarda les coordenades globals en el fitxer designat per l'usuari, amb extensió .tris.

El format DIMUC

Justificació

Amb l'explicació anterior sobre l'exportació de coordenades del format .tris es posa de manifest un petit problema: el format .tris està orientat a treballar amb malles independents, deixant al programador la responsabilitat de col·locar-les adequadament a l'espai 3D a les aplicacions basades en RPQ.

A vegades cal poder col·locar les malles de forma relativa a d'altres. Imaginem el cas de voler mostrar una representació 3D del cos humà on tenim una malla per cada òrgan. La tasca d'haver de compondre l'escena “a ma” dintre l'aplicació pot ser força tediosa. La forma més simple seria muntar el model dintre del programa de modelat, centrar-lo i exportar la informació de cada malla juntament amb la seva matriu de transformació. Per aquest motiu es va dissenyar el format .DIMUC i les seves eines derivades.

Aquestes eines tenen com a finalitat generar models .tris de manera còmode al dissenyador.

Especificació

El format DIMUC es basa en fitxers .DIMUC i en fitxers .gm, els fitxers .gm contenen la informació relativa a cada malla i el fitxer .DIMUC guarda el factor d'escala i el llistat de fitxers .gm que conté

l'escena (figura A1.3). S'ha de tenir en compte que el nom del fitxer .gm és el nom que tenia l'objecte a *Blender*.

```
#version:
2.0
100.0
home1.gm
home2.gm
home3.gm
home4.gm
home5.gm
home6.gm
```

Figura A1.3: Extracte d'un arxiu .DIMUC

Un arxiu .gm conté la informació relativa a una malla, això són la llista dels seus vèrtexs, la llista de cares com a 3 indexes a la llista de vèrtexs, la matriu de transformació, les normals per vèrtexs i la informació de la textura de la malla (figura A1.4).

```
# numero de caras
3442

# matriz de transformacion
0.253010898829 0.0 0.0 0.0
0.0 0.253010898829 0.0 0.0
0.0 0.0 0.253010898829 0.0
0.0925737023354 0.1182474792 0.496186316013 1.0

# numero de vertices
1939

# lista de vertices
-2.68606901169 -1.60053503513 -8.06737327576
-2.66920804977 -1.59254896641 -8.06283664703
-2.66561603546 -1.6157130003 -8.04846763611
-2.68247699738 -1.62369894981 -8.05300235748
-2.73770499229 -1.54260504246 -7.88369607925
-2.72165608406 -1.53073501587 -8.01955413818
```



```
-2.74980592728 -1.55004394054 -8.02776527405
```

```
...
```

```
# lista de caras
```

```
3 0 2
```

```
0 1 2
```

```
7 4 5
```

```
5 6 7
```

```
9 10 11
```

```
8 9 11
```

```
12 13 3
```

```
3 2 12
```

```
9 14 15
```

```
16 6 5
```

```
...
```

Figura A1.4: Extracte d'un arxiu .gm

Importador/Exportador .DIMUC

L'importador/exportador de fitxers .DIMUC salva els problemes abans esmentats ja que permet importar i exportar escenes complexes sense pèrdua d'informació rellevant.

El conversor dimuc2tris

El format DIMUC no pot ser carregat directament per RPQ, per això es fa servir un conversor: dimuc2tris. Aquest conversor genera els fitxers .tris de l'escena continguda en un fitxer .DIMUC, a part permet certes opcions (figura A1.5) com són exportar coordenades globals o locals, convertir diverses malles a una de sola, generar fitxer de normals per vèrtex o generar fitxer de materials, entre d'altres.

```
USO: dimuc2tris [OPCIONES] fichero .DIMUC
```

```
    dimuc2tris --help
```

OPCIONES:

--no-wt Solo se tiene en cuenta las coordenadas locales de los
vertices

--one-mesh Todas las mallas se colapsan en una

--normals Genera un fichero de normales por vertice: solo a partir
de la version 2

--materials Genera un fichero de materiales de la malla: solo a partir
de la version 3

--help Escribe esta ayuda

Figura A1.5: sortida per pantalla de l'ajuda de dimuc2tris

Novetats de la versió 2.0

En aquesta versió es va afegir suport per generar fitxers .nrml amb el conversor dimuc2tris. Aquest fitxer conté les normals per vèrtex en el mateix ordre en que apareix el seu corresponent vèrtex en el fitxer .tris.

Aquesta informació es fa servir per generar una visualització més realista d'objectes polits, oferint un sistema senzill de suavitzat de superfície sense haver d'incrementar el nombre de triangles de la malla.

Per activar la generació de fitxers de normals s'ha de fer servir la opció "--normals". El fitxer generat té el mateix nom que el fitxer .tris de la malla en qüestió però amb l'extensió .nrml.

Novetats de la versió 3.0

La novetat principal d'aquesta versió és la exportació de materials des de *Blender* a *.DIMUC* i, posteriorment obtenir un fitxer *.mtrl* amb la informació del material de la malla associada.

La informació, guardada per línies a aquest fitxer és la següent:

4. Color ambient i difós (R, G, B).
5. Component alfa de transparència.
6. Color especular (R, G, B).
7. Transparència del color especular.
8. Coeficient d'especularitat.

Per activar la generació del fitxer de materials es fa servir la opció “—materials” de *dimuc2tris*. El fitxer generat té el mateix nom que el fitxer de la malla associada però amb l'extensió *.mtrl*.

Instal·lació

Els *scripts* es reconeixen per tenir una extensió *.py* i s'han de copiar a una carpeta de *Blender* per tenir-los disponibles a la secció d'importar/exportar del menú principal (figura A1.6). S'han de copiar a:

*C:\Archivos de programa\Blender
Foundation\Blender\blender\scripts*

El programa *dimuc2tris* és un executable de línia de comandes que no fa servir cap llibreria addicional.

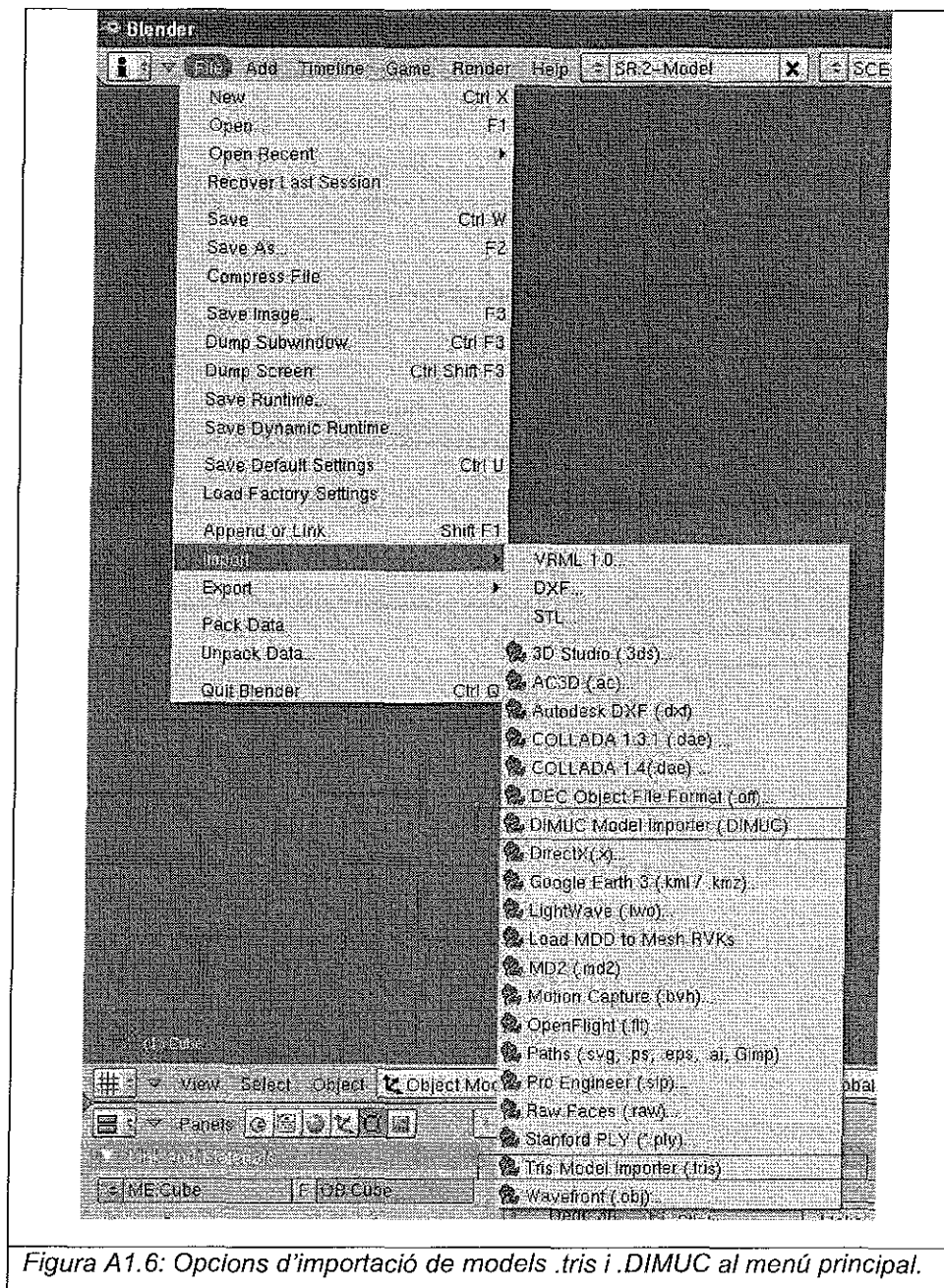


Figura A1.6: Opcions d'importació de models .tris i .DIMUC al menú principal.

Versions:

- Versió 3.0 publicada el 30/05/2008
 - Suport para la exportació de materials per malla (.mtrl).
 - Correcció de bugs.
- Versió 2.0 publicada el 7/05/2008
 - Suport per exportació de normals per vèrtex(.nrml).
 - importació dels noms de malles.

- Versió 1.0 publicada el 17/01/2008
 - Importació/exportació des de *blender* a .DIMUC
 - Conversor .DIMUC->.tris
 - Importador/exportador .tris

Annex II: Manual del software de control del robot pòrtic d'un braç

Introducció:

El projecte d'investigació de nous procediments quirúrgics per operacions amb laparoscòpia es pot dividir en dos grans branques:

1. Control i seguiment automàtic de la càmera al camp de visió òptim del cirurgià
2. Captura i reproducció dels moviments del cirurgià dintre del pacient.

Totes dues branques tenen el mateix procediment:

1. Simular mitjançant un entorn virtual el comportament del robot proposat.
2. Validar l'arquitectura del robot proposat i modificar paràmetres, si convé, per poder cobrir les necessitats mèdiques reals.
3. Exportar la intel·ligència artificial al robot real.

La primera part s'ha implementat completament i actualment consta d'un software simulador/controlador d'un robot pòrtic d'un braç. A continuació s'adjunta un breu manual de funcionament del simulador en la seva primera *Release Candidate*.

L'entorn de simulació

Aquest entorn es divideix en dos parts diferenciades: el formulari de control y les vistes 3D (figura A2.1)

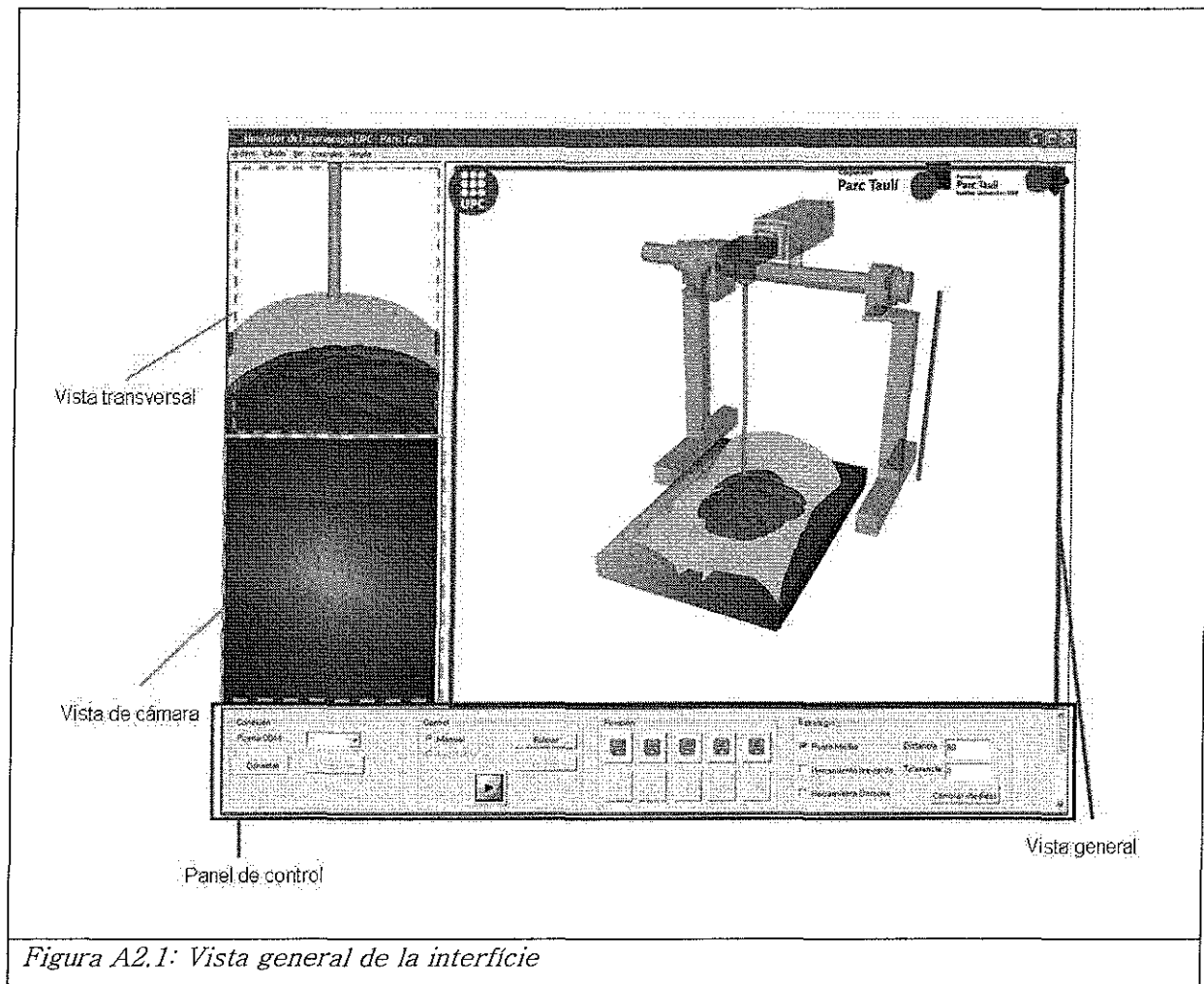


Figura A2.1: Vista general de la interfície

Les opcions del formulari que es mostren inicialment són les opcions bàsiques de control. Si fem més gran El formulari podem veure la resta d'opcions.

Les noves opcions inclouen operacions sobre la cinemàtica directa

i inversa del robot i la seva utilitat és purament conceptual i de proves (figura A2.2).

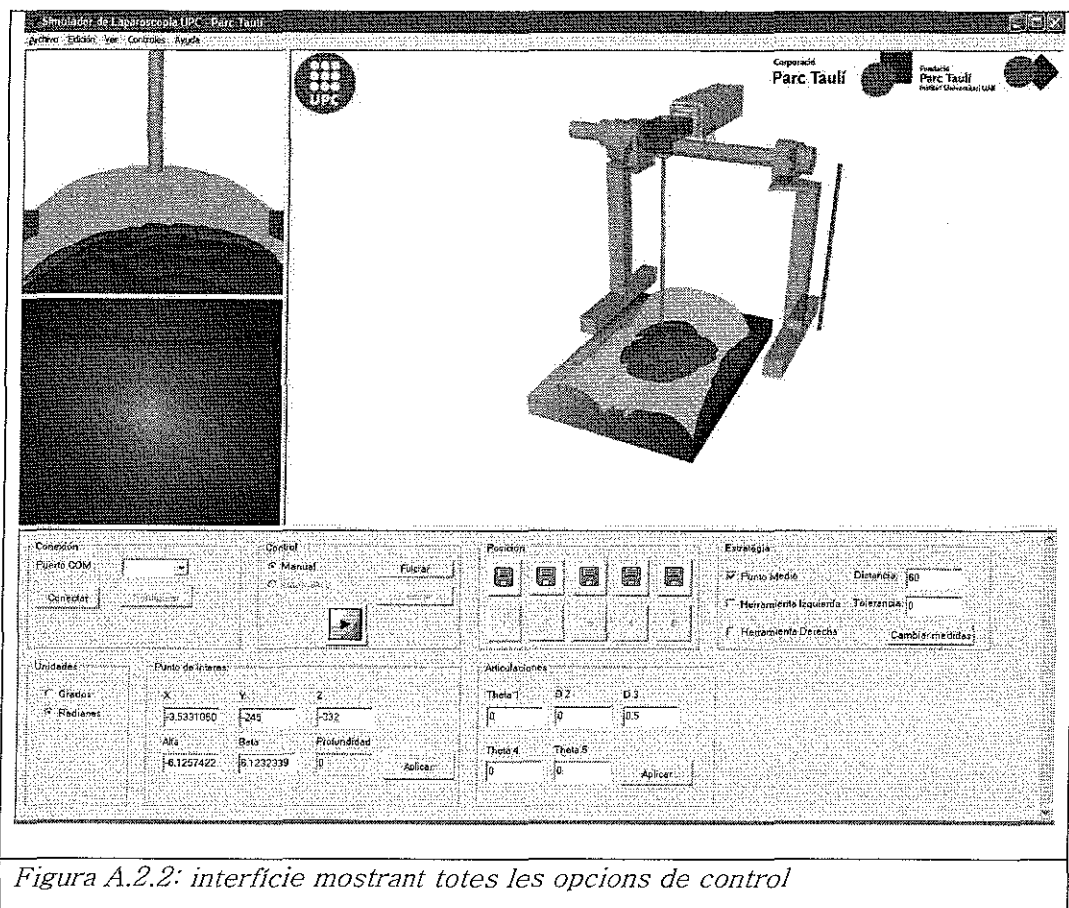


Figura A.2.2: interfície mostrant totes les opcions de control

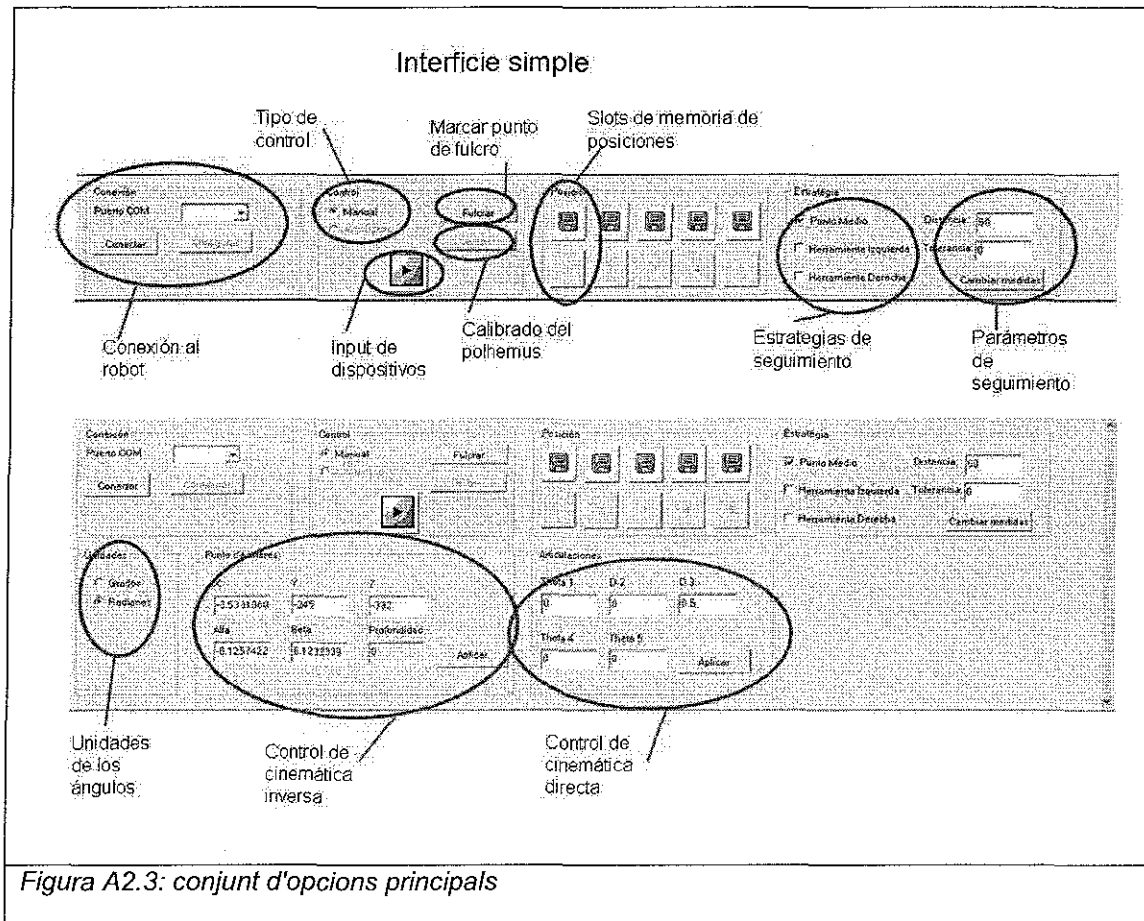
Operacions principals:

Les opcions principals del simulador s'agrupen en 6 grups principals (figura A2.3):

1. Connexió amb el robot.
2. Input del robot i comportament.
3. Cal·libració dels sensors de posició.
4. Memorització i reproducció de posicions específiques.
5. Configuració de l'estratègia de seguiment.
6. Cinemàtica:

6.1. Cinemàtica Directa

6.2.Cinemàtica inversa



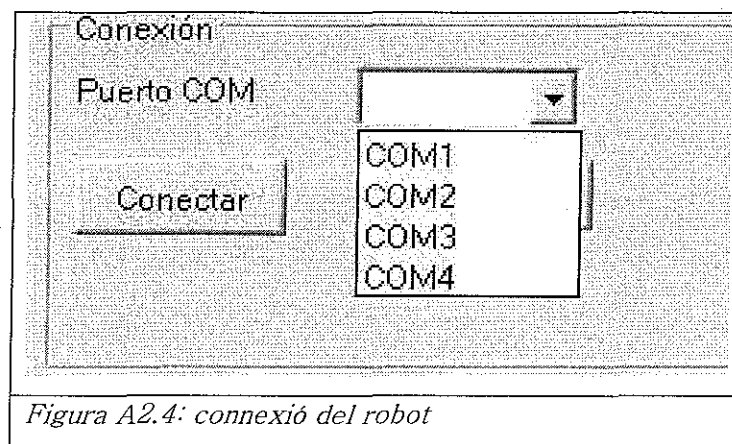
Connexió amb el robot

Quan connectem amb el robot tots els moviments que fem amb el simulador també es traduiran al robot. La comunicació amb la controladora es fa mitjançant una connexió sèrie seguint el protocol especificat.

AVIS IMPORTANT!! quan es connecta el robot aquest es posiciona segons la configuració del simulador. Tècnicament hauria de ser a la inversa: quan es connectes el robot es el simulador qui pregunta la configuració actual d'aquest i la mostra. El problema es que la

retroalimentació de posició està parcialment implementada a la controladora i, de moment no es fa servir.

Per connectar amb el robot s'ha d'escollir el port en el desplegable y pitxar el botó *Conectar* (figura A2.4).



Input del robot i comportament

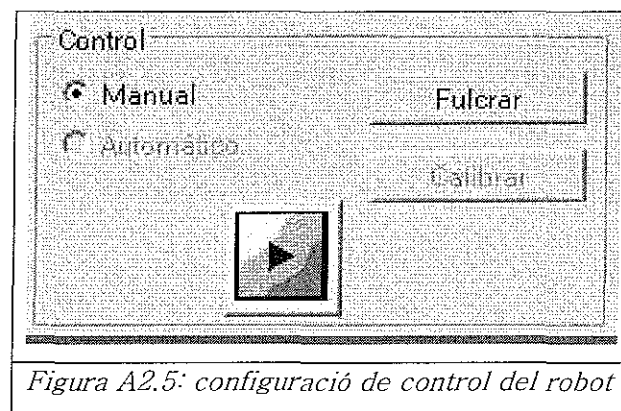
El robot pot ésser controlat de forma manual i automàtica. L'entrada de dades del control es pot habilitar i deshabilitar amb el botó



. Amb aquest botó es posa en marxa o es frena el *thread* d'enquesta de dispositiu.

Si es vol governar manualment es pot fer servir el teclat o un joystick. Quan es controla de forma automàtica aquest es posiciona en funció dels sensors de posició que hi han a les eines. El mode automàtic només es pot fer servir un cop el robot s'ha fulcrat. Per una operació normal el procediment seria el següent:

1. Guiar el robot fins al punt de fulcre de forma manual (amb teclat o joystick).
2. Fulcrar. Això s'aconsegueix pressionant el botó *Fulcrar* quan el robot es al punt desitjat.
3. Introduir les eines dintre del robot y cal·librar-les.
4. Passar a mode automàtic (pitjant l'opció *Automático*) (figura A2.5).
5. Si fa falta ajustar l'estratègia i els seus paràmetres.



Cal·libració dels sensors de posició

Per defecte tots els càlculs que es fan al simulador es calculen definint un sistema de coordenades centrat en el robot. Això genera un problema a l'hora de treballar amb sistemes de posicionament magnètics externs (*Polhemus*) i és que aquests donen les posicions segons el seu sistema de coordenades que generalment està centrat en el seu emissor de camp. Cal, doncs transformar les coordenades del sensor a coordenades del robot. Per això s'ha de trobar una matriu M tal que, donat un punt P_s del sistema de coordenades del robot el transformi en un punt P_b del sistema de referència del robot.

$$P_b = M \times P_s$$

$$\begin{bmatrix} Pb_x \\ Pb_y \\ Pb_z \\ 1 \end{bmatrix} = \begin{bmatrix} & & T_x \\ & R & T_y \\ & & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} Ps_x \\ Ps_y \\ Ps_z \\ 1 \end{bmatrix}$$

Si $R=I$ l'equació es simplifica ja que:

$$Pb = Ps + T$$

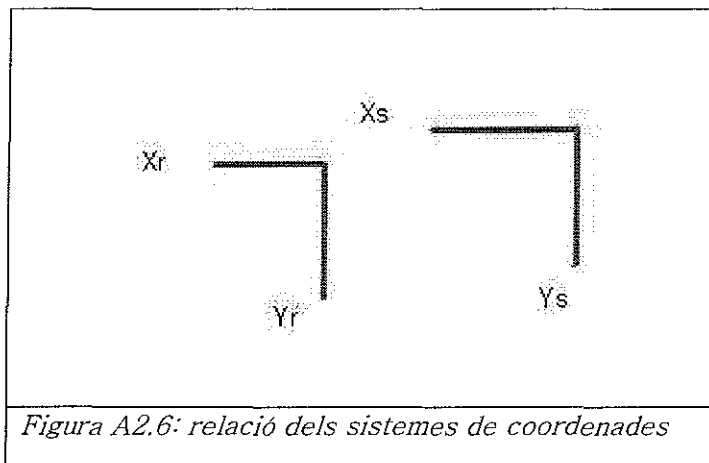
La cal·libració inicial serveix per establir T . Això es fa intentant obtenir les coordenades d'un punt P arbitrari en el sistema de coordenades del robot (Pb) i en el sistema de coordenades del sensor (Ps).

Aquest sensors s'han configurat per ésser a la part superior d'eines de 42.5 centímetres. Actualment si es vol canviar la longitud de les eines s'ha d'anar al fitxer `ControlEscena.cpp` y canviar la macro `LTOOL` pel valor adient.

El procés de cal·libració es du a terme quan s'ha fulcrat el robot. en aquest moment s'introdueixen les eines dintre del pacient, es pitxa el botó *Calibrar* i es col·loquen les eines davant la càmera, tal i com demana el missatge que s'obre, a la distància marcada al camp *Distancia* (per defecte 6 centímetres). Un cop centrades les eines dins el camp de la càmera es prem el botó *Aceptar* del missatge. En aquest moment Pb és el punt que està a la distància marcada segons l'eix de la càmera i Ps és el punt mig del segment que uneix els dos extrems de les eines. Ja tenim $T = Pb - Ps$.

Aquest procediment té una pega: els sistemes sensors de posició no mapegen tot l'espai sinó només el semiespai positiu de les X . Aquest

problema es pot resoldre col·locant el sensor de tal forma que tot el robot quedi situat dintre d'aquest semiespai i fent $R = I$ (figura A2.6).



El problema principal és l'error generat: les mostres que es prenen més allunyades de l'emissor de camp del *Polhemus* (O_s) tenen molt més error. Això fa que les mostres entre un sensor i l'altre siguin molt dispars.

La solució passa per col·locar el sistema dels sensors de forma que mapegi la zona de treball dels sensors i, a més, la distància dels sensors a l'origen del seu sistema de referència sigui molt semblant (figura A2.7).

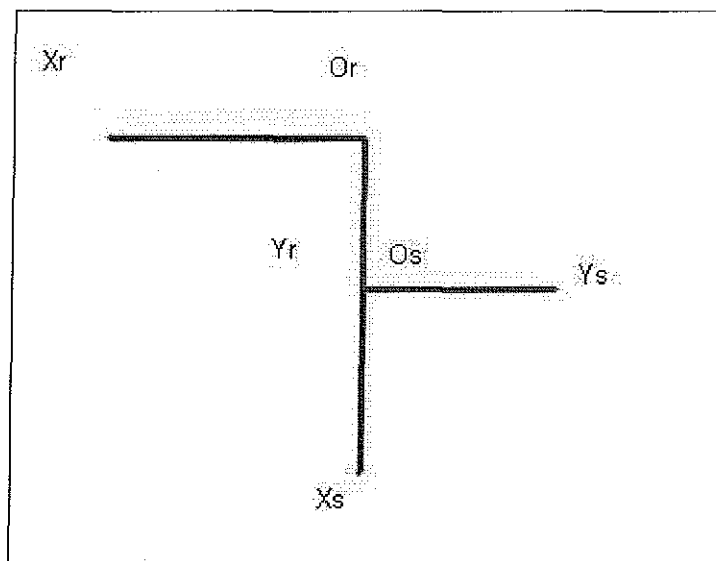


Figura A2.7: relació dels sistemes de coordenades idònia

Això, a priori fa que R no sigui I si hem de mantenir T tal i com l'hem definit fins ara. Com que la transformació d'eixos és trivial podem donar una $T' = f(T)$ que fa que $R = I$

$$\begin{bmatrix} & & T_x \\ & R & T_y \\ & & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} & & T'_x \\ & I & T'_y \\ & & T'_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La transformació es la següent:

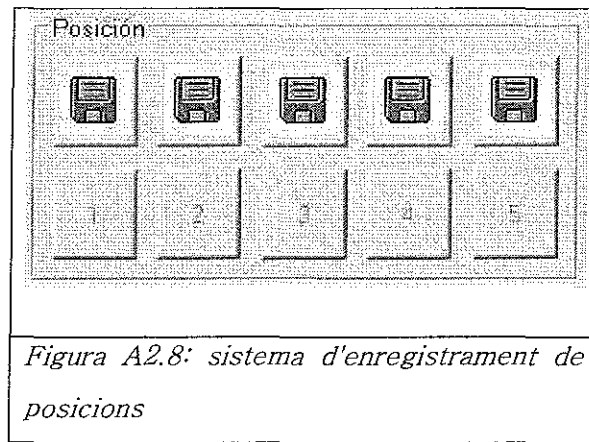
$$\begin{aligned} T'_x &= T_y \\ T'_y &= -T_x \\ T'_z &= T_z \end{aligned}$$


De forma que

$$Pb = Ps + T'$$

Memorització i reproducció de posicions específiques

El sistema permet enregistrar posicions del robot i tornar a anar a una d'elles només pressionant un botó (figura A2.8). Per això es disposa d'un màxim de 5 *slots* per guardar i sobreescriure posicions. A més, cada cop que es marca una nova posició aquesta es mostra a la interfície per tenir una referència visual clara. S'ha de dir que el que es guarda és la configuració del robot que situa la càmera a aquell punt.



Cada cop que es prem un dels botons  s'enregistra la posició actual i es guarda a l'slot corresponent. Per anar a aquella posició només s'ha de pressionar el botó amb el número d'slot corresponent.

Configuració de l'estratègia de seguiment

La posició de la càmera es determina a partir de la posició de les eines. La posició de les eines determina un punt d'interès (PDI). La càmera es configura per col·locar-se a una distància D d'aquest punt (sempre i quant el fet de mantenir aquesta distància no impliqui treure la càmera del pacient). A més es pot definir un radi de tolerància, H per fer que les eines es puguin moure dintre d'una regió sense que la càmera es mogui (figura A2.9):

Si $H > \text{dist}(PDI_i, PDI_{ref})$ llavors

$\text{moureCamera}(PDI_i, Fc, D, Ld)$

$PDI_{ref} = PDI_i$

Fsi

Estratègia	
<input checked="" type="checkbox"/> Punto Medio	Distancia: <input type="text" value="60"/>
<input type="checkbox"/> Herramienta Izquierda	Tolerancia: <input type="text" value="0"/>
<input type="checkbox"/> Herramienta Derecha	<input type="button" value="Cambiar medidas"/>

Figura A2.9: configuració de l'estratègia de seguiment

El PDI a seguir pot ser l'extrem de l'eina esquerra, l'extrem de l'eina dreta o el punt mig del segment que uneix els dos extrems de les eines.

Cinemàtica

Les opcions de cinemàtica són opcions avançades que es van dissenyar per fer proves de comportament i controls d'errors. Es poden fer proves de cinemàtica directa o inversa (figura A2.10).

Punto de interés:			Articulaciones		
X	Y	Z	Theta 1	D 2	D 3
<input type="text" value="-3.5331060"/>	<input type="text" value="-245"/>	<input type="text" value="-332"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0.5"/>
Alfa	Beta	Profundidad	Theta 4	Theta 5	
<input type="text" value="-6.1257422"/>	<input type="text" value="6.1232339"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	
<input type="button" value="Aplicar"/>			<input type="button" value="Aplicar"/>		

Figura A2.10: control de cinemática

Cinemàtica Directa:

La cinemàtica directa és aquella que resol el sistema de coordenades de l'extrem del robot dient els valors que prenen les seves articulacions.

Aquest problema es resol gràcies a fer servir la notació de Denavit-Hartenberg.

Cinemàtica Inversa:

La cinemàtica inversa resol el problema de saber quins valors han de tenir les articulacions d'un robot perquè l'extrem del robot es col·loqui a un punt determinat de l'espai. En molts casos la solució no és única i el problema es complica bastant. Una forma de destriar possibilitats és establir un sistema de coordenades objectiu en comptes d'un punt. O sigui, donar els valors de les articulacions per que el sistema de coordenades de l'extrem del robot coincideixi amb un sistema donat.

Per resoldre aquest problemes hi han varies vies. L'estudi dels jacobians permet trobar solucions genèriques per qualsevol robot lineal amb un cost computacional més alt i menor precisió. Altres formes genèriques serien per *Hill-Climbing* a partir d'un mapa de solucions donat. En aquest robot s'ha optat per una solució purament geomètrica i particular. Per més informació veure l'explicació de la cinemàtica inversa.

El simulador en qüestió resol la cinemàtica en funció de si està fulcrada o no. Si no ho està es resol la cinemàtica per un punt i els angles d'Euler de l'eina. En cas d'estar fulcrada es resol la cinemàtica per una profunditat donada (considerant el punt de fulcre) i els angles d'Euler.

Dreceres de teclat

Certes funcionalitats es poden fer directament a través de teclat en comptes de prémer un botó de la interfície. A continuació ve el llistat de dreceres:

Funcionalitat	Drecera
Fulcrar l'eina	f
Alliberar l'eina	espai
Guardar posició a l'slot <i>i</i>	Shift + num <i>i</i>
Anar a posició <i>i</i> -essima	Ctrl + num <i>i</i>
Seguiment de l'eina esquerra	i
Seguiment de l'eina dreta	d
Seguiment del punt mig de les eines	m
Incrementar coordenada X de l'extrem	fletxa LEFT
Decrementar coordenada X de l'extrem	fletxa RIGHT
Incrementar coordenada Y de l'extrem	fletxa DOWN
Decrementar coordenada Y de l'extrem	fletxa UP
Incrementar coordenada Z de l'extrem	Re. Pag
Decrementar coordenada Z de l'extrem	Av. Pag
Incrementar alfa	z
Decrementar alfa	x

Incrementar beta	c
Decrementar beta	v
Activar/desactivar entrada per dispositiu	s

Annex III: Plànols del prototip

